

GABRIEL ANGELO NAZÁRIO

Orientadora: Andréa Iabrudi Tavares

**SIMULAÇÃO DE COLETA DE DADOS EM REDES DE
SENSORES SEM FIO POR ROBÔS MÓVEIS UTILIZANDO
A FERRAMENTA PLAYER STAGE**

Ouro Preto

Novembro de 2012

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**SIMULAÇÃO DE COLETA DE DADOS EM REDES DE
SENSORES SEM FIO POR ROBÔS MÓVEIS UTILIZANDO
A FERRAMENTA PLAYER STAGE**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

GABRIEL ANGELO NAZÁRIO

Ouro Preto
Novembro de 2012



UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

Simulação de Coleta de Dados em Redes de Sensores Sem Fio por Robôs
Móveis Utilizando a Ferramenta Player Stage

GABRIEL ANGELO NAZÁRIO

Monografia defendida e aprovada pela banca examinadora constituída por:

Dra. ANDRÉA IABRUDI TAVARES – Orientadora
Universidade Federal de Ouro Preto

Dr. HAROLDO GAMBINI SANTOS
Universidade Federal de Ouro Preto

Dr. RICARDO AUGUSTO RABELO OLIVEIRA
Universidade Federal de Ouro Preto

Ouro Preto, Novembro de 2012

Resumo

Em uma rede de sensores sem fio onde a coleta de dados será feita com o auxílio de robôs móveis, o planejamento das heurísticas de roteamento é uma tarefa complexa de se realizar por vários motivos: incerteza do robô sobre a posição dos nós sensores, falta de conhecimento do fenômeno sendo monitorado, características estocásticas presentes em uma situação real, entre outros. Perdas de dados - que ocorrem quando um nó sensor esgota sua capacidade de armazenamento e, assim, deixa de coletar mais dados - devem ser evitadas ao máximo. Além da perda de dados, outro aspecto importante que deve ser observado é o consumo de energia pelos nós sensores. Além de ser de difícil resolução, é improvável que seja encontrada uma solução exata para esse problema. Para minimizar a perda de dados, é desejável que algumas heurísticas sejam avaliadas através de simulação em diferentes cenários e configurações de rede.

Este trabalho avalia quantitativamente o impacto de se considerar a taxa de aquisição de dados de cada nó sensor nas heurísticas para o planejamento da rota, em diferentes cenários de disposição espacial, capacidade de armazenamento e taxa de aquisição. Dentre as quatro heurísticas abordadas nesse trabalho, uma delas considera em conjunto com a taxa de aquisição de dados o raio de comunicação dos nós sensores (vizinhança), sendo essa heurística uma abordagem totalmente nova.

Os resultados obtidos nesse trabalho são provenientes de simulações realizadas nas ferramentas Player e Stage. Essas duas ferramentas foram escolhidas pois permitem que vários cenários de teste com características muito próximas da realidade sejam utilizados. O Player fornece uma interface de rede para comunicação entre o código e o *hardware* dos robôs, muito utilizado em robôs reais. Para simular o robô e o ambiente onde ele se encontra, juntamente com os nós sensores, é utilizado o Stage, que toma o lugar do *hardware* do robô.

Com a avaliação dos resultados obtidos através das simulações, pôde-se observar uma diminuição significativa da perda de dados quando se considera a taxa de aquisição de dados dos nós sensores durante o planejamento da rota. Essa diminuição foi ainda melhorada quando se utilizou a aquisição de dados em conjunto com a vizinhança entre os nós sensores.

Palavras-chave: Coleta de dados. Redes de sensores sem fio. Robôs móveis.

Abstract

In a wireless sensor network where data collection will be done with the help of mobile robots, route planning is a complex task to accomplish for several reasons: robot's uncertainty about the position of sensor nodes, lack of knowledge of the phenomenon being monitored, stochastic features present in a real situation, among others. Loss of data - that occur when a sensor node exhausts its storage capacity, and thus fails to collect more data - should be avoided as much as possible. Another important aspect that should be noted is the energy consumption by sensor nodes. Besides being difficult to resolve, is unlikely to be found an exact solution to this problem. To minimize data loss, it is desirable that some heuristics are evaluated by simulation in different scenarios and network configurations.

This work quantitatively evaluates the impact of considering the rate of data acquisition on each sensor node on heuristics for route planning in different settings of spatial distribution, storage capacity and acquisition rate. Among the four heuristics discussed in this work, one of them considers together with the rate of data acquisition communication, the radius of sensor nodes (neighborhood), which is a completely new approach.

The results obtained in this study come from simulations performed in the Player and Stage tools. These two tools were chosen because they allow several test scenarios with characteristics very close to reality are used. The Player provides a network interface for communication between the code and the robot's hardware, widely used in real robots. To simulate the robot and the environment in which he finds himself, along with the sensor nodes, the Stage is used.

With the evaluation of the results obtained through simulations, we observed a significant reduction in data loss when one considers the rate of data acquisition of sensor nodes during route planning. This decrease was further enhanced when using the data acquisition together with the neighborhood between the sensor nodes.

Keywords: Data collection. Wireless sensor networks. Mobile robots.

*“Mesmo que houvesse um exército contra mim, meu coração não temeria.” - Payable On
Death*

Agradecimentos

Agradeço a Deus, por guiar meus passos. Ao meu anjo da guarda, pela proteção. Aos meus pais, pelos exemplos, ensinamentos, força e por acreditarem sempre em mim. Familiares e amigos, pelo incentivo e companheirismo. À República Ninho do Amor, por todo apoio e aprendizado durante esse tempo em Ouro Preto. Aos meus professores, em especial à minha orientadora, pela confiança, oportunidade e paciência.

Sumário

1	Introdução	1
2	Justificativa e Trabalhos Relacionados	6
3	Metodologia	9
3.1	Modelo	9
3.2	Heurísticas	10
4	Desenvolvimento	16
4.1	As ferramentas Player e Stage	16
4.2	Código	18
4.3	Cenários	19
5	Experimentos e Resultados	22
5.1	Resultados	23
6	Conclusões	27
	Referências Bibliográficas	42

Lista de Figuras

1.1	Exemplos e componentes básicos de nós sensores	3
1.2	Capacidade de comunicação dos nós sensores	4
1.3	Possíveis maneiras de se coletar dados em redes de sensores sem fio	5
3.1	Exemplo de tomada de decisão sobre qual nó sensor deve ser o próximo a ser visitado, de acordo com a segunda heurística	13
3.2	Exemplo genérico de otimização sobre uma rota gerada	15
4.1	Modelos de funcionamento das ferramentas Player e Stage	17
4.2	Modelo usado para as simulações	17
4.3	Diagrama de classes usadas no trabalho.	18
4.4	<i>Screenshots</i> capturadas durante uma simulação	20
5.1	Gráficos apresentando as médias de tempo e perda de dados para cada um dos quatro tipos de carga.	26

Lista de Tabelas

5.1	Tabela comparativa de heurísticas.	23
-----	--	----

Capítulo 1

Introdução

O objetivo desse trabalho é analisar o impacto de se utilizar informações sobre a aquisição de dados dos nós sensores durante o planejamento da rota para a coleta desses dados por robôs móveis em redes de sensores sem fio. Através dessa análise, é possível avaliar em quais cenários há diminuição da perda de dados.

As Redes de Sensores Sem Fio (RSSF) são cada vez mais utilizadas, pois são opções simples e baratas para o monitoramento e sensoriamento de fenômenos, especialmente em ambientes inóspitos ou de difícil acesso. Uma grande variedade de aplicações pode utilizar esse tipo de rede em sua estrutura [1, 7], como por exemplo:

- Meio ambiente: monitoramento de variáveis ambientais (tais como calor, umidade e temperatura, etc) em prédios, casas, florestas, oceanos, entre outros.
- Militar: monitoramento, rastreamento, segurança, principalmente em áreas perigosas e/ou de difícil acesso.
- Tráfego: monitoramento de vias, controle de tráfego.

O uso desse tipo de tecnologia é algo relativamente novo, e se fez possível graças ao avanço no desenvolvimento de circuitos integrados, micro-processadores e padrões de comunicação sem fio. Uma rede de sensores sem fio é constituída de:

- Observador: usuário ou sistema computacional (aplicação), que deseja obter respostas sobre o fenômeno monitorado.
- Nó sensor: responsável pela monitoração do fenômeno através de medidas locais e por passar as informações ao observador, que depois irá estudá-las.

A arquitetura de um nó sensor é ilustrada na Figura 1.1d, onde também são apresentadas algumas fotografias de nós sensores reais, mostrando sua variabilidade. Cada nó sensor da rede coleta localmente dados sobre o fenômeno de interesse, através de sensores internos. Existem

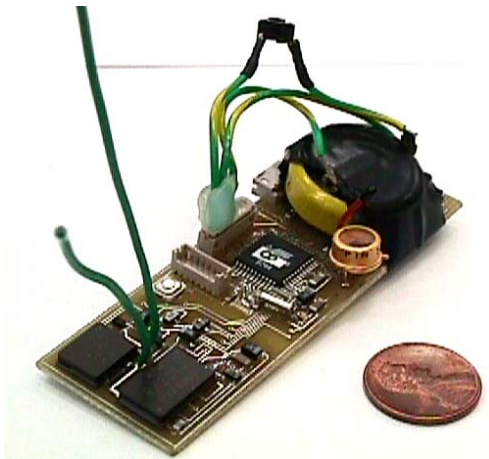
sensores para medir temperatura, umidade, calor, acústica, infravermelho, entre outros. Os dados coletados podem ser processados e armazenados localmente, sendo a capacidade de armazenamento definida pela memória do elemento. Em algum momento, o nó sensor deve enviar seus dados locais para o observador. Para isso, ele utiliza seu transceptor, que possui um raio de comunicação onde seu sinal pode ser alcançado. Cada uma destas operações consome energia de sua bateria, sendo a mais cara a comunicação [3]. Como em geral a bateria não é recarregável, o consumo de energia é questão central em RSSFs.

O modelo abstrato para comunicação em uma RSSF é ilustrado na Figura 1.2. Cada nó tem, de acordo com suas especificações técnicas, um raio de comunicação. Os nós podem ter diferentes tamanhos de raios, ou seja, diferentes capacidades de comunicação. Além disso, esse raio pode ser afetado por obstáculos (como uma rocha) ou por interferência causada por outras ondas atuando no mesmo ambiente, alterando a forma circular da região. Um nó sensor pode se comunicar com outro elemento se houver interseção entre suas áreas de comunicação.

Dependendo do fenômeno de interesse monitorado, a taxa de aquisição de dados amostrados pela rede é diferente em cada nó. Em fenômenos difusos, que acontecem igualmente em toda região monitorada, como por exemplo, umidade, essa taxa de aquisição tende a ser homogênea entre os nós. Já em fenômenos concentrados, alguns nós possuem taxa de aquisição de dados superior a outros e a rede é denominada heterogênea. Um exemplo é uma rede que monitora a ocorrência de incêndios em uma floresta [16]: caso algum nó colete indício de evento anormal (como um pequeno aumento de temperatura), deve entrar em estado de “alerta” e coletar dados a uma frequência maior, verificando se realmente o incêndio está ocorrendo. Como a memória em um sensor é limitada, quanto maior a taxa de aquisição, mais rapidamente essa memória irá esgotar, antes do envio dos dados locais para o observador.

A forma mais comum de transmissão de dados para o observador é a multi-salto (ou *multi-hop*), onde os dados trafegam pela rede passando de nó em nó, até o nó central (sorvedouro). A comunicação consome muita energia [3], já que cada nó sensor deve receber os dados dos nós vizinhos e retransmití-los. Por exemplo, com o roteamento estabelecido na Figura 1.3a, alguns nós transmitem poucas quantidades de dados (Nó 1), enquanto que outros recebem um grande volume de dados (Nós 2 e 3). A energia consumida pela forma multi-salto pode ser muito grande e concentrada, levando à expiração do recurso. Com um ou mais nós sensores fora de operação (seja por falta de memória ou energia), o tempo de vida da rede diminui.

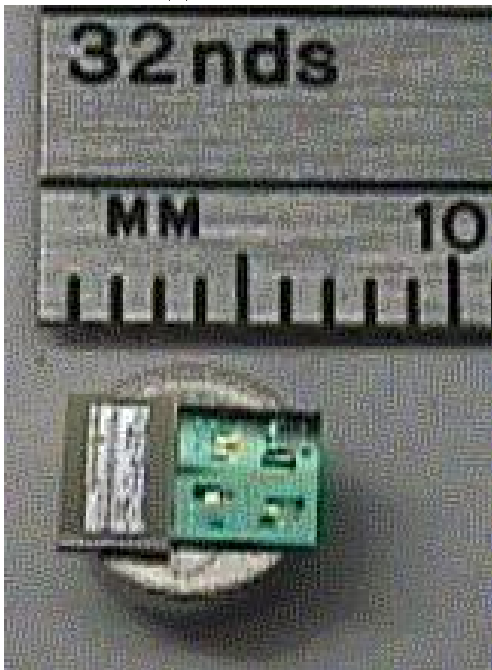
Uma alternativa para diminuir o consumo de energia em RSSFs é utilizar um robô móvel que percorre o ambiente coletando os dados armazenados nos nós sensores [3, 2]. Nessa abordagem, os dados são coletados quando o robô atinge a área de comunicação do nó sensor, como mostrado na Figura 1.3b, evitando a transmissão dos dados de forma multi-salto. Com a ajuda dos robôs, há economia da energia dispendida em comunicação na rede, aumentando seu tempo de vida. O uso de robôs móveis permite ainda a coleta de dados em RSSFs que sejam total ou parcialmente esparsas (onde um ou mais nós sensores são desconexos dos demais), ou



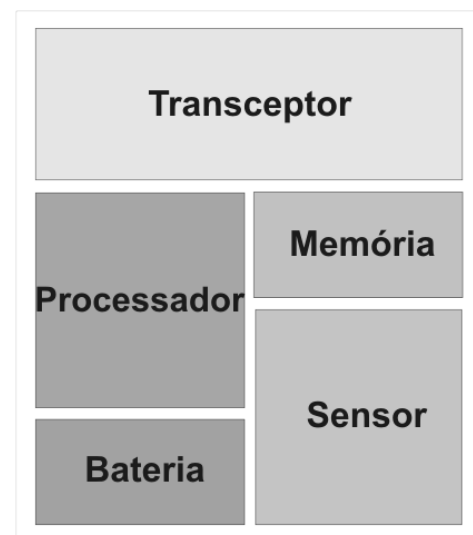
(a) COTS Dust.



(b) Mica2.



(c) SMART Dust.



(d) Componentes básicos de um nó sensor.

Figura 1.1: Exemplos de nós sensores: COTS Dust (Figura 1.1a), Mica2 (Figura 1.1b) e SMART Dust (Figura 1.1c), todos desenvolvidos pela *University of California, Berkeley*. A Figura 1.1d ilustra os componentes básicos de um nó sensor: transceptor (componente que combina um transmissor e receptor), processador, bateria, memória e sensor.

seja, a forma multi-salto não poderia ser utilizada pois não há conectividade entre todos os nós sensores.

Um dos fatores determinantes no sucesso da coleta por robôs móveis é a definição da rota de coleta, ou seja, da ordem em que os nós sensores serão visitados, o que induz uma trajetória para o robô (como mostrado pela linha hachurada da Figura 1.3b). A rota pode causar perda de dados devido ao esgotamento da memória do nó sensor antes da visita pelo

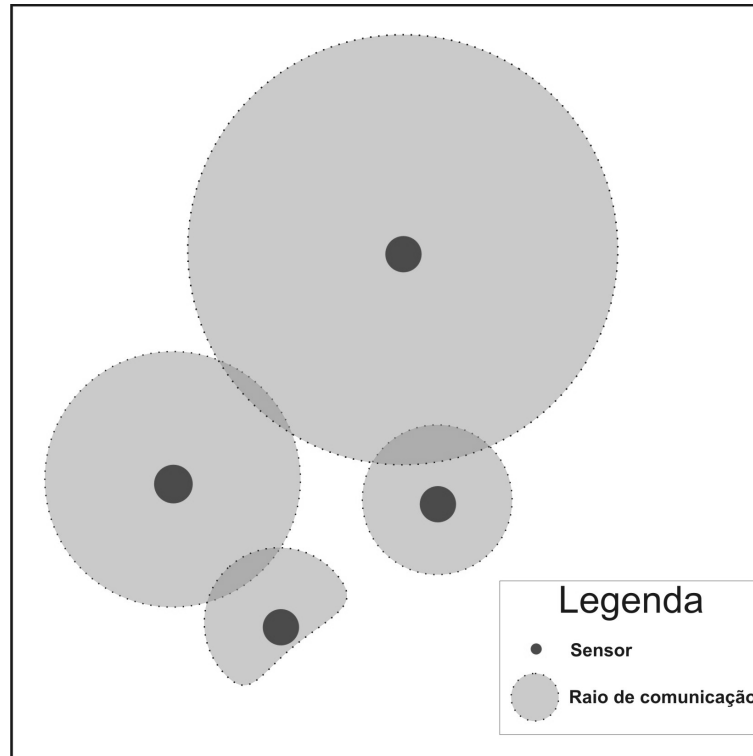
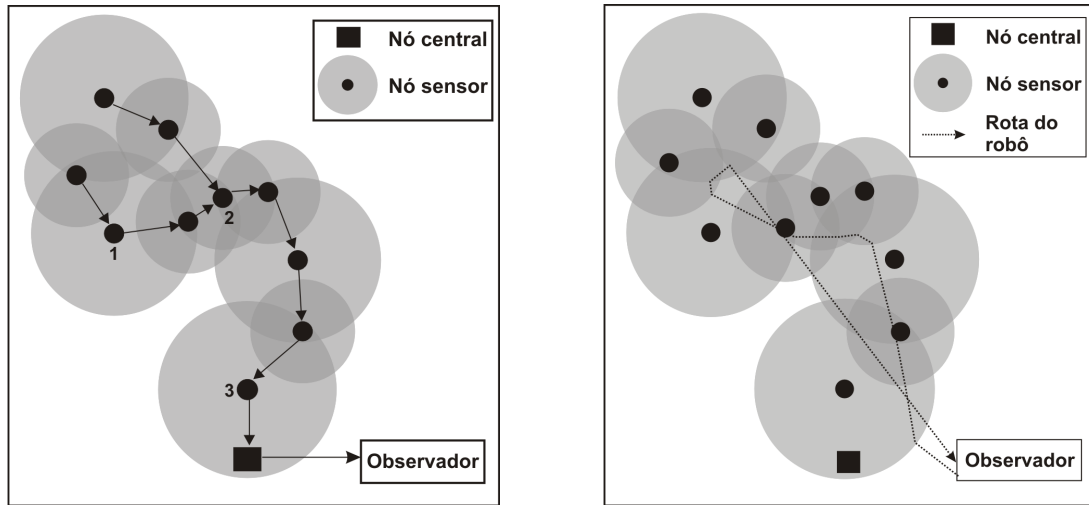


Figura 1.2: Representação esquemática das capacidades de comunicação de uma RSSF.

robô. Outro aspecto importante, porém potencialmente conflitante, é a energia consumida pelo robô, que é determinada principalmente pelo custo de navegação na execução da rota e, em geral, proporcional à distância percorrida [10].

O objetivo desse trabalho é analisar o impacto de se utilizar informações sobre a aquisição de dados dos nós sensores durante o planejamento da rota para a coleta desses dados por robôs móveis em redes de sensores sem fio. Pela complexidade do problema, foram propostas diferentes heurísticas que tratam de aspectos complementares do mesmo. Essas heurísticas de roteamento foram utilizadas em diversos cenários, permitindo que seja feita uma análise dessas heurísticas sobre o roteamento do robô. Foram comparados os resultados onde essa taxa de aquisição é levada em conta com os resultados onde esse parâmetro não é considerado. Para essa comparação, foram realizadas simulações da coleta de dados utilizando o Player Stage, um ambiente de simulação muito utilizado em robótica. Outro objetivo é definir um modelo no Player Stage para a realização das simulações. Esse modelo será útil não apenas para esse trabalho, mas poderá ser usado em outros projetos dentro do contexto das redes de sensores sem fio.

O restante do texto está organizado da seguinte forma: no Capítulo 2 são apresentadas as razões de se realizar esse trabalho, mostrando sua importância e dificuldades envolvidas na sua realização, bem como são revisados os principais trabalhos relacionados. O Capítulo 3



(a) Envio dos dados dos sensores de forma multi-hop.

(b) Coleta dos dados com ajuda de robôs.

Figura 1.3: Possíveis maneiras de se coletar dados em redes de sensores sem fio. Através da abordagem multi-salto (Figura 1.3a), os dados são passados de um nó sensor ao outro, até chegarem ao nó central. Outra forma de coleta utiliza um robô móvel, que percorre o ambiente coletando os dados dos nós sensores (Figura 1.3b).

apresenta a metodologia, com as etapas e abordagens utilizadas para a realização do trabalho, assim como as heurísticas de roteamento do robô. O Capítulo 4 discute aspectos da implementação, incluindo a apresentação do ambiente de simulação. O Capítulo 5 descreve como foram realizados os experimentos, com os cenários de teste utilizados para as simulações, e apresenta os resultados obtidos, com uma análise comparativa das heurísticas. O Capítulo 6 finaliza com as principais conclusões, ressaltando as contribuições realizadas e indicando trabalhos futuros.

Capítulo 2

Justificativa e Trabalhos Relacionados

Redes de sensores sem fio se tornaram alternativas viáveis para uma grande variedade de aplicações, como por exemplo monitoramento de ambientes, medidas de segurança e detecção de eventos. Uma RSSF pode conter milhares de nós sensores, dependendo do ambiente e da aplicação selecionada. Todo dado coletado pelo sensor deve ser enviado a uma estação base, para posterior processamento desses dados coletados [1]. Como discutido anteriormente, para que esses dados sejam entregues à base, é usual uma transmissão multi-salto, de um nó para outro, o que implica em total conexão da rede para que a base seja atingível por todos os nós sensores. Além disso, há um gasto grande e desbalanceado de energia para comunicação. Diante desses fatos, surgiu a idéia de se desenvolver RSSF com elementos móveis [2], que podem ser os próprios nós sensores, as estações base ou algum outro elemento especial que auxilie na coleta. Isso permite a implantação de redes esparsas, onde nem todos os nós sensores precisam estar conectados aos demais [3].

Neste trabalho, o elemento móvel se trata de um robô que percorre o ambiente onde os nós sensores se encontram espalhados para efetivar a coleta. Com o robô visitando e coletando os dados, a forma multi-salto é evitada. O planejamento da rota do robô no ambiente é uma parte essencial desse tipo de rede, estando intimamente ligada à perda de dados. Esse problema vem sendo abordado sob diversas perspectivas diferentes [3], visando aumentar tempo de vida da rede, diminuir latência de entrega de dados na base, aumentar a cobertura, evitar obstáculos, além dos aspectos de coleta colaborativa.

O controle da trajetória do robô que percorre o ambiente pode ser estática ou dinâmica. Uma trajetória estática indica um caminhar que não muda com o tempo, ou seja, para cada vez que o robô faz a coleta, o caminho percorrido é sempre o mesmo, bem como a ordem das visitas. Uma trajetória dinâmica indica um caminhar que pode mudar a ordem das visitas até mesmo durante a execução da coleta. Várias soluções apresentam uma trajetória estática para os elementos móveis, como proposto neste trabalho. Essas soluções dependem da distribuição da rede e do número de elementos móveis presentes. Em [8], os nós sensores são distribuídos de forma circular e densa, resultando em uma trajetória em círculo utilizando

apenas um elemento móvel. Já [4] propõe vários elementos móveis que se direcionam a uma quantidade limitada de áreas dentro da RSSF. O problema de planejamento do caminhamento é tratado em [15], onde um único elemento móvel (ou *data mule*) visita todos os nós sensores presentes na rede, podendo ser esparsa ou não. Mesmo que a rede seja densa, cada nó sensor deve passar seus dados diretamente ao elemento móvel, de forma *single-hop*.

Durante a coleta dos dados, um ponto importante a ser observado é o tempo gasto para a transferência dos dados entre o nó sensor e o elemento móvel. Esse tempo, além de depender da distância entre eles, depende também se o elemento móvel está em movimento ou parado para a coleta [5, 14]. Neste trabalho, a velocidade do robô é sempre constante, e o processo de transferência de dados é considerado instantâneo.

Em um cenário hipotético ideal, os sensores teriam uma capacidade de armazenamento infinita, ou seja, independentemente da sua taxa de aquisição de dados, não haveria esgotamento da memória. Assim, a trajetória realizada pelo robô não faria diferença na perda, pois em qualquer ordem de visitas, todos os nós sensores seriam alcançados sem ocorrer perda de dados. Mesmo quando somente a economia da energia dispendida pelo robô é considerada para definição da rota, o problema é NP-difícil, já que é uma versão Problema do Caixeiro Viajante[6]. No contexto da rota de robôs móveis para coleta de dados em RSSF, deve-se observar ainda a perda de dados, além de vários aspectos de incerteza, como, por exemplo, a posição exata de um nó sensor, ou a presença de um obstáculo que impeça o robô de alcançar a área de comunicação (raio de atuação) desse nó. São diversas as abordagens na literatura para definição de rota com consideração de perda de dados, indo desde uma política quase reativa [11] até planejamento sofisticado levando em conta a latência [15]. Em [10], os aspectos estocásticos são modelados e integrados num arcabouço de tomada de decisão para definição da rota, mas não são consideradas heterogeneidades na aquisição e capacidade de armazenamento.

Este trabalho ignora os aspectos estocásticos, mas considera que os nós sensores possuem diferentes taxas de aquisição e memória limitada e avalia a perda de dados de acordo com as características do fenômeno monitorado. A taxa de aquisição de dados já foi considerada anteriormente [2, 13], onde o problema foi modelado como um Caixeiro Viajante com Janela de Tempo [12]. Segundo [2], esse problema pode ser classificado como uma variação do problema do Caixeiro Viajante, denominado ATSP_{TW} (*Asymmetric Traveling Salesman Problem with Time-Windows*), onde cada nó sensor tem uma janela de tempo onde pode a coleta deve ser realizada, antes que sua capacidade de armazenamento se esgote. Após esse tempo, haverá perda de dados.

Mais recentemente, várias abordagens utilizam o modelo de Caixeiro Viajante com Vizinhos para representar a possibilidade de se comunicar com vários nós sensores em um único ponto [17, 15]. Esse problema é classificado por [15] como TSPN (*Traveling Salesman Problem with Neighborhoods*). A grande novidade deste trabalho se encontra na heurística utilizada

para resolver a combinação entre ATSP_{TW} e TSP_N.

Na literatura são frequentes trabalhos que enfoquem principalmente a distância percorrida pelo robô, embora alguns abordem explicitamente a diminuição de perda e latência dos dados. Não se encontrou, entretanto, uma avaliação do impacto de diferentes tipos fenômenos – difusos ou concentrados – na eficácia da rota, sendo este o principal foco deste trabalho. A complexidade do problema e a diversidade de abordagens existentes na literatura justificam a necessidade de uma avaliação comparativa de heurísticas diferentes com relação a perda de dados. Ambientes simulados são largamente utilizados em robótica e RSSFs, para experimentação de diferentes cenários. Essa é a abordagem utilizada neste trabalho: a comparação de diferentes heurísticas para definição de rotas para coleta, através da sua aplicação em diversas instâncias do problema, com desempenho medido em ambiente simulado.

Capítulo 3

Metodologia

A avaliação do volume de perda de dados em heurísticas de definição de rota que levem em conta a aquisição de dados é o foco principal deste trabalho e a metodologia adotada é a comparação de indicadores obtidos pelas mesmas em ambientes simulados e sob diferentes cenários.

O uso de simulação se mostra uma alternativa inicial mais viável do que a realização de experimentos reais, devido ao custo e complexidade de preparação dos mesmos. A definição dos diversos cenários de teste são importantes para que se possa avaliar empiricamente o desempenho das heurísticas. Através das simulações, além da facilidade de se preparar esses cenários, o tempo exigido para a realização dos testes é muito menor que o teste real. As heurísticas foram avaliadas utilizando-se simulações nas ferramentas Player e Stage (<http://playerstage.sourceforge.net/>), ambiente bastante utilizado pela comunidade de Robótica.

Mesmo em ambientes simulados, é necessário incluir mais algumas simplificações sobre o problema geral. O estabelecimento do problema resolvido e do modelo é abordado na primeira seção deste capítulo. Em seguida, são descritas as heurísticas escolhidas para serem comparadas. O capítulo termina com uma discussão sobre os cenários utilizados para comparação, com especial ênfase diferentes fenômenos (difusos e concentrados) capturados pela caracterização das taxas de aquisição.

3.1 Modelo

O modelo proposto captura os aspectos relevantes do problema real para investigação em questão. Neste trabalho, devido a limitações de tempo e complexidade, alguns detalhes foram simplificados. Em uma RSSF real, os nós sensores estão espalhados num ambiente, que pode sofrer ação de agentes externos. Há a incerteza do robô sobre sua própria posição e a posição dos nós sensores no ambiente. O robô deve estimar, durante toda sua trajetória, sua própria posição e a posição dos nós sensores utilizando dados imprecisos que consegue capturar

(informações de sensores internos). Há ainda a incerteza sobre os fenômenos que estão sendo monitorados pelos nós sensores, o que influencia diretamente na taxa de aquisição de dados. Um robô real deve saber lidar com essa incerteza para a tomada de decisões que levem ao resultado esperado. No problema simplificado, não haverá agentes externos que modifiquem a estrutura do ambiente e essas incertezas não serão consideradas.

Além disso, são simplificados alguns aspectos do ambiente real. O ambiente simulado é um quadrado de tamanho fixo e é abstratamente visualizado como um grid, composto por diversas células. Além disso, no ambiente só estão presentes o robô e os nós sensores, isto é, o robô não deve se preocupar em verificar se há ou não obstáculos no seu caminho durante a coleta. Dessa forma, e considerando-se a velocidade do robô constante, há uma proporção direta entre a distância euclidiana e o tempo de navegação.

Dados n nós sensores, cada nó i é caracterizado por sua posição fixa (x_i, y_i) , seu raio de comunicação r_i e seu limite de armazenamento b_i . O fenômeno sob supervisão é caracterizado pelo vetor n -dimensional \mathbf{c} da carga de dados induzida na posição de cada nó-sensor c_i por unidade de tempo. A carga é considerada constante durante o período de coleta.

O robô tem uma posição (x_r, y_r) e uma velocidade constante v e possui conhecimento sobre todos parâmetros do sistema descritos acima. O planejamento de rota consiste em definir a ordem na qual os n nós serão visitados para coleta de dados. A distância percorrida pelo robô ao implementar uma rota é considerada como a soma entre as distâncias entre os nós sucessivos mais a distância inicial do robô até o primeiro nó. Esse valor é um limite superior para a distância real percorrida, já que os dados de um nó sensor i são coletados instantaneamente assim que o robô atinja sua área de comunicação, ou seja, quando a distância entre ambos seja menor ou igual a r_i , não sendo necessário atingir a posição exata do nó-sensor (veja exemplo na Figura 3.2a).

3.2 Heurísticas

Diferentes abordagens para definição de rota são propostas, simuladas e analisadas, utilizando como principais indicadores a perda de dados e a distância percorrida pelo robô. Quando a única medida quantitativa da qualidade da rota de coleta é a distância percorrida pelo robô, ou seja, deseja-se apenas minimizar a energia consumida pelo robô sem preocupação com perda de dados, o problema é modelado como um Caixeiro Viajante [6] clássico, onde a matriz de distâncias é obtida pela distância entre os nós-sensores. Há diversas soluções exatas, aproximadas e heurísticas para o problema que podem ser utilizadas. Quando além da distância, deseja-se minimizar a perda de dados, o problema se aproxima da formulação do Caixeiro Viajante com janelas de tempo [12], onde o nó i tem janela que varia entre 0 e $\lceil \frac{b_i}{c_i} \rceil$, para que seja evitada perda. Mais recentemente, incorporou-se à modelagem do problema a observação de que um nó-sensor pode ter seus dados coletados, mesmo que esse nó não seja o

próximo da rota do robô, desde que este último tenha entrado em sua área de comunicação. Esse problema é um caso especial do Problema do Caixeiro Viajante com Vizinhanças tratado por [15]. A primeira heurística é um algoritmo guloso para o problema do Caixeiro Viajante[6], considerando apenas as distâncias entre os nós sensores, utilizada como base de comparação. A segunda heurística é um algoritmo guloso para o problema do Caixeiro Viajante com Janela de Tempo[12], considerando além distâncias entre os nós, as taxas de aquisição de dados de cada um deles e a capacidade de memória para definir as janelas de visita. A terceira heurística é uma solução para o problema do Caixeiro Viajante com Vizinhanças[17, 15], ou seja, o robô pode coletar os dados de um nó sensor, mesmo que ele não seja o próximo a ser visitado, desde que sua área de comunicação intercepte a trajetória do robô. Finalmente, a quarta heurística faz uma combinação entre a segunda e a terceira, utilizando como parâmetros para a definição da rota a aquisição de dados e a vizinhança. É importante destacar que essa abordagem que utiliza em conjunto a taxa de aquisição de dados e vizinhança entre nós sensores é nova, e se mostrou mais eficiente que as demais em vários cenários, avaliando-se a perda de dados. Para todas as heurísticas são considerados indicadores de desempenho a distância total percorrida pelo robô e o volume de dados que são perdidos.

As heurísticas são comparadas através de resultados de simulação, onde é avaliado o impacto da utilização das informações da taxa de aquisição de dados dos nós sensores e a utilização da vizinhança durante o planejamento da rota do robô. Para realizar essa comparação, foram utilizadas como métricas de desempenho o tempo total gasto pelo robô para coletar os dados de todos os nós sensores e a perda total de dados.

A primeira heurística trata-se de uma heurística gulosa simples para o Problema do Caixeiro Viajante e servirá como base de comparação para o desempenho das outras abordagens. Essa heurística utiliza um algoritmo simples no cálculo da rota, onde o próximo sensor a ser visitado é sempre o que está a menor distância da posição atual. É utilizada somente a matriz de distâncias entre o robô e os sensores, e as distâncias entre cada um dos sensores. O pseudo-código deste algoritmo guloso está disponível no Algoritmo 1).

A segunda heurística é uma solução gulosa para Problema do Caixeiro Viajante com Janela de Tempo, que leva em conta a memória e taxa de aquisição dos nós sensores. Ela prioriza a visita àqueles nós cujo esgotamento de memória está mais próximo. Cada sensor tem uma janela de tempo onde pode operar sem que ocorra a perda de dados. Para considerá-la, é calculado quanto tempo ainda resta até o estouro da memória e o tempo que será gasto pelo robô até alcançá-lo. A menor diferença entre esses dois tempos indica qual sensor deverá ser escolhido para ser visitado. Caso essa diferença de tempos resulte em empate entre dois ou mais nós sensores, será escolhido aquele que tem o menor tempo até que sua memória acabe. Um exemplo da decisão tomada pelo algoritmo pode ser visto na Figura 3.1. O Algoritmo 2 mostra como é feita essa escolha, até que todos os sensores tenham seus dados coletados.

A terceira heurística é uma solução gulosa para Problema do Caixeiro Viajante com Vi-

Algoritmo 1: Primeira heurística de roteamento, onde o robô não vai considerar as taxas de dados produzidos pelos sensores, apenas as distâncias entre eles e o robô.

entrada: Lista s contendo os n sensores a serem visitados.

saída : Vetor r com a ordem das visitas.

```

1 x = coordenada x do robô;
2 y = coordenada y do robô;
3 for i = 0 to n do
4     escolher o sensor presente em s que está à menor distância de (x, y);
5     inserir o índice do sensor escolhido em r;
6     x = coordenada x do sensor escolhido;
7     y = coordenada y do sensor escolhido;
8     remover o sensor escolhido de s;
9 end
10 retornar r;
```

Algoritmo 2: Segunda heurística de roteamento. Nessa heurística, para cada sensor ainda não visitado, deverá ser calculado o tempo restante até que ele comece a perder dados, e quanto tempo o robô levará para alcançá-lo. A menor diferença entre esses tempos indica qual o sensor deverá ser o próximo a ter seus dados coletados.

entrada: Lista s contendo os n sensores.

saída : Vetor r com a ordem das visitas.

```

1 x = coordenada x do robô;
2 y = coordenada y do robô;
3 for i = 0 to n do
4     selecionar o sensor que possui a menor diferença entre o tempo gasto pelo robô
      para alcançá-lo e tempo até o estouro do buffer. Caso exista mais de um sensor
      com a mesma diferença (empate), selecionar o mais próximo das coordenadas x
      e y;
5     inserir o índice do sensor escolhido em r;
6     x = coordenada x do sensor escolhido - raio do sensor;
7     y = coordenada y do sensor escolhido - raio do sensor;
8     remover o sensor escolhido de s;
9 end
10 retornar r;
```

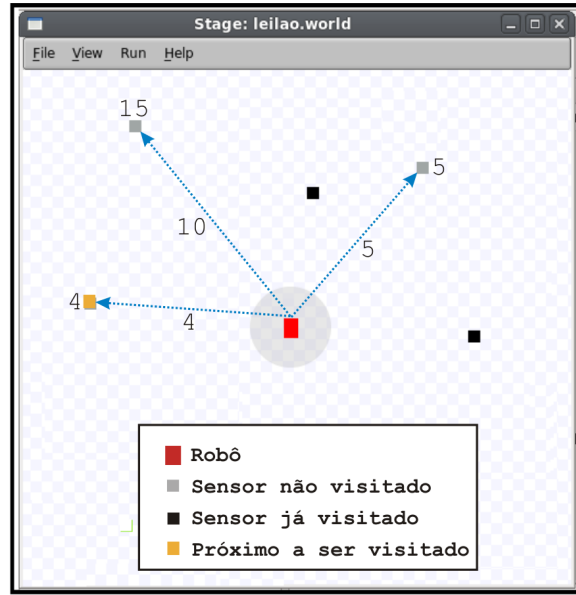


Figura 3.1: As setas azuis indicam a rota a ser seguida e o tempo que será gasto para que o robô alcance o próximo sensor a ser visitado. Os números ao lado do sensor indicam quanto tempo resta até que ocorra o estouro de memória. De acordo com o segundo algoritmo, o sensor escolhido será aquele que tiver a menor diferença entre esses tempos. Em situação de empate (como no caso dessa figura), será escolhido o sensor que tiver menos tempo até que sua memória se esgote.

zinhanças. Nesta versão do problema, é montado um grafo com todas as possíveis arestas de caminhamento do robô (desde que passe pelo raio de comunicação dos nós sensores), além daquelas da rota original. O ciclo procurado não precisa conter todos os nós do grafo, mas pelo menos um vizinho de todos os nós do grafo. Esse modelo é bastante apropriado para a coleta em RSSFs, pois não há necessidade de se visitar exatamente a posição do nó sensor para realizar a coleta, bastando entrar no raio de comunicação desse nó.

A heurística proposta recebe como entrada uma rota qualquer gerada e a otimiza, encontrando o menor caminho para o robô considerando agora a vizinhança dos nós sensores. Isso significa que ao percorrer o caminho entre dois sensores o robô pode atravessar o raio de comunicação de um terceiro (ou mais) sensor. Dessa forma, a coleta pode ser feita nesse mesmo instante, e esse terceiro sensor pode ser retirado da rota original, pois teve seus dados coletados sem a necessidade de que o robô caminhasse exclusivamente para ele, diminuindo assim a distância percorrida pelo robô.

O modelo de otimização de rota é exemplificado na Figura 3.2. Definida a rota inicial, o grafo de vizinhança é montado da seguinte forma. Se no caminho entre um nó sensor i e outro nó sensor j , que não é o imediatamente subsequente a i na rota, o robô passa pela área de comunicação de todos os sensores $(i + 1, \dots, j)$, então uma aresta (i, j) é adicionada ao grafo,

Algoritmo 3: Descrever algoritmo 3.

entrada: Vetor r com a ordem de visitas aos n sensores.
saida : Vetor s com a ordem das visitas otimizada, considerando a vizinhança entre eles.

```

1 for  $i = 0$  to  $n - 1$  do
2    $j = i + 1$ ;
3    $inseriu = true$ ;
4   while  $inseriu$   $j \leq n$  do
5     constrói segmento de reta  $l_{ij}$ ;
6      $k = i + 1$ ;
7     while  $inseriu$   $k < j$  do
8       calcula a distância  $d$  entre  $l_{ij}$  e sensor  $m_k$ ;
9        $inseriu = (d \leq \text{raio do sensor } m_k)$ ;
10       $k = k + 1$ ;
11    end
12    if  $inseriu$  then
13      insere aresta  $(i, j)$  no grafo  $G$ ;
14       $j = j + 1$ ;
15    end
16  end
17 end
18  $s =$  menor caminho no grafo  $G$  pelo algoritmo de Dijkstra;
19 retornar  $s$ ;

```

com o custo da aresta sendo a distância entre esses dois nós. A aresta representa que os dados dos nós sensores intermediários podem ser coletados enquanto o robô navega de i para j , sem necessidade de desvio.

A Figura 3.2 ilustra o processo de montagem do grafo de vizinhança a partir de uma rota pré-estabelecida. Dado um conjunto de nós sensores, seja a rota gerada (1-2-3-4-5-6) e a navegação correspondente mostrada na Figura 3.2a. O grafo de vizinhança é construído a partir dos caminhos possíveis entre o robô e os nós sensores (Figura 3.2b). Caso o robô possa se comunicar com todos os nós intermediários (sua trajetória cruza o círculo de comunicação) entre dois nós sensores, uma aresta é inserida no grafo de vizinhança. Para o exemplo em questão, o grafo de vizinhança resultante é apresentado na Figura 3.2c. Por exemplo, em vez de seguir do ponto inicial (p) para o nó sensor 1, e em seguida para o nó sensor 2, é possível seguir do ponto inicial p diretamente para o nó sensor 2, pois nesse trajeto o robô passa pelo raio de comunicação do nó sensor 1. Porém, não é possível passar do ponto inicial p para o nó sensor 3, pois nesse trajeto (p-3), o robô não passa pelos nós sensores anteriores na rota previamente definida. Esse procedimento deve ser feito com todos os nós sensores subsequentes, até que se chegue ao último, ou que não seja possível passar pelo raio de atuação de todos os nós sensores anteriores. Depois de montado o grafo, o algoritmo de Dijkstra é

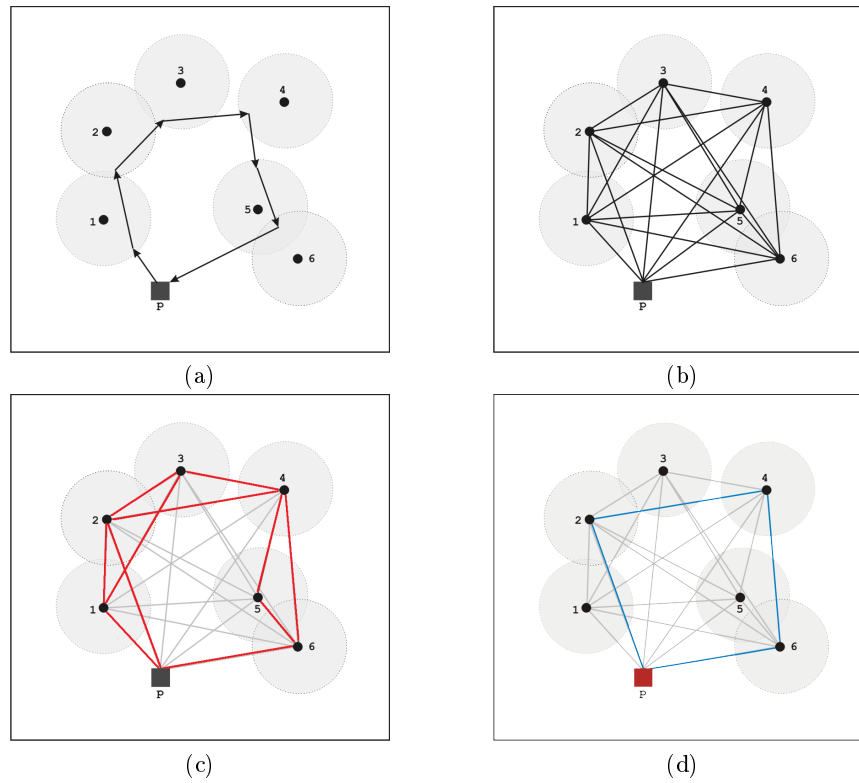


Figura 3.2: Exemplo genérico de otimização sobre uma rota gerada. Inicialmente, é gerada a rota de visitas por alguma das heurísticas apresentadas anteriormente (Figura ?? e Figura 3.2a). Em seguida, é montado um grafo, adicionando arestas que possam eliminar algumas visitas, desde que o robô passe pelo raio de comunicação do nó sensor. Depois do grafo pronto, o algoritmo de menor caminho de Dijkstra é utilizado, para que o robô possa percorrer a rota otimizada.

utilizado para se encontrar o menor caminho no mesmo. Os nós sensores que participam do menor caminho formam a nova rota. No exemplo, a rota otimizada é ilustrada na Figura 3.2d.

Finalmente, a quarta heurística é uma combinação entre a segunda (que considera as taxas de aquisição de dados de cada nó sensor) e a terceira: dada a rota gerada pelo segundo planejamento, será encontrado o menor caminho considerando agora a vizinhança dos nós sensores. Logo, essa quarta heurística considera para a definição da rota a taxa de aquisição de dados e a vizinhança de cada sensor. Esta proposta de integrar as duas heurísticas é nova e como será mostrado, supera as anteriores tanto na distância percorrida quanto na diminuição de perda de dados.

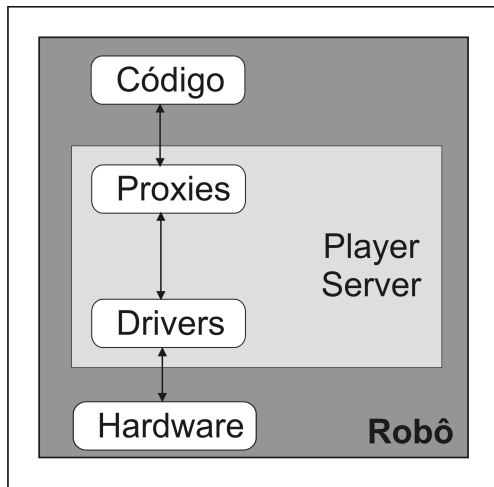
Capítulo 4

Desenvolvimento

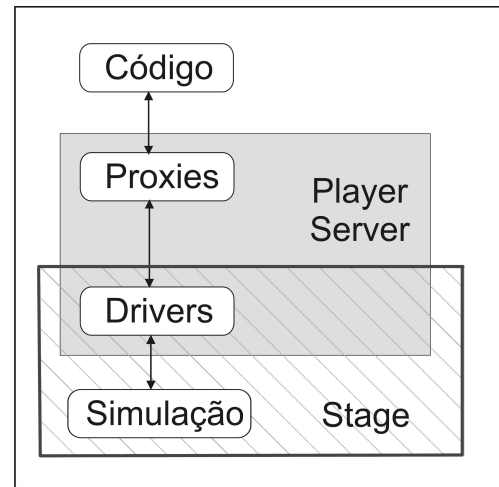
4.1 As ferramentas Player e Stage

As ferramentas Player e Stage serão utilizadas para a realização das simulações. A página *web* do projeto (<http://playerstage.sourceforge.net/>) fornece informações sobre as ferramentas e funcionamento. O Player fornece uma interface de rede para uma grande variedade de *hardware* de robôs e sensores reais. O modelo cliente/servidor permite que os programas de controle do robô possam ser escritos em várias linguagens de programação (entre elas Java, C, C++ e Python) e executados em qualquer computador com uma conexão de rede para o robô. O Stage é um simulador de múltiplos robôs que simula uma população de robôs móveis que se deslocam em um ambiente em duas dimensões. Vários modelos de sensores são fornecidos para serem utilizados na simulação. Os dispositivos presentes no Stage apresentam uma interface padrão com o Player, o que permite que os programas de controle usados na simulação possam ser transferidos para o *hardware* de um robô real, necessitando de poucas ou nenhuma alteração em seu código. Vários controladores projetados no Stage demonstraram trabalhar perfeitamente em robôs reais. A Figura 4.1a mostra a estrutura cliente/servidor do Player quando usado em um robô real. A Figura 4.1b mostra a estrutura do Player/Stage sendo usados em um ambiente de simulação.

O ambiente onde os elementos (nós sensores e robô) se encontram será referenciado como mundo: um *grid* de tamanho fixo, como mostrado na Figura 4.2. O modelo que será usado nesse trabalho consiste de nós sensores espalhados pelo mundo, cada um em uma célula do *grid*. Vale lembrar que esses elementos são estáticos, ou seja, não sofrem alterações tais como mudanças na posição (o que pode ocorrer em um ambiente real, mas que não será tratado nesse trabalho). O robô deve percorrer o mundo em direção aos nós sensores para coletar seus dados. A movimentação se dá com o robô navegando pelas células do *grid*, passando para células vizinhas.



(a) Player sendo usado em um robô real.



(b) Player/Stage sendo usados em uma simulação.

Figura 4.1: Modelos de funcionamento das ferramentas Player e Stage. A Figura 4.1a mostra a estrutura de comunicação cliente/servidor entre o Player e um robô real. A Figura 4.1b mostra a interação entre Player e Stage usados em uma simulação.

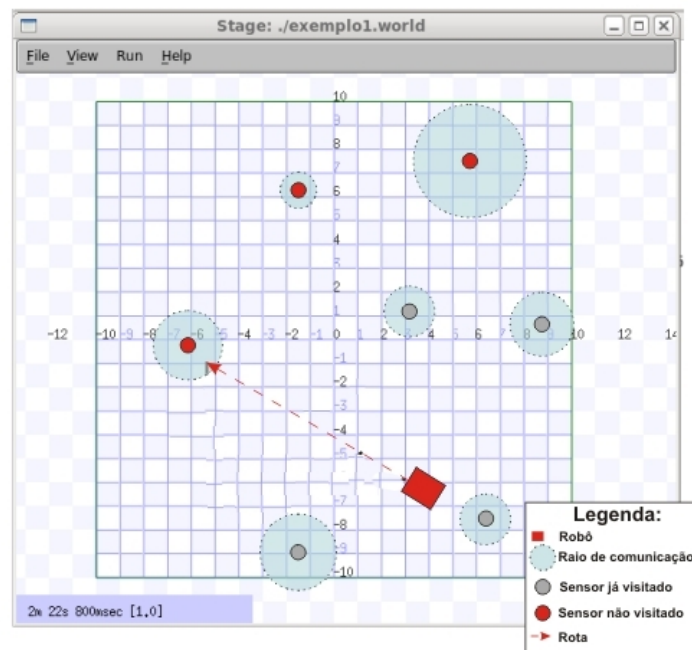


Figura 4.2: Modelo usado para as simulações. Consiste de um *grid* de tamanho fixo, com os elementos (nós sensores e robô móvel) distribuídos pelo seu espaço.

4.2 Código

Para a realização desse trabalho, foi feito uso do código de autoria de Fabrício Feitosa Carvalho, afiliado ao VerLab - Laboratório de Visão Computacional e Robótica (<http://www.verlab.dcc.ufmg.br/>), do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais. Através desse código, escrito em C++, são criados todos os elementos necessários para a simulação: o mundo (classe Map), o robô (classe Robot) e sensores (classe Target). A primeira, segunda, terceira e quarta heurísticas foram implementadas nos métodos Planejamento1, Planejamento2, Planejamento3 e Planejamento4 respectivamente. A Figura 4.3 mostra um diagrama de classes do código utilizado.

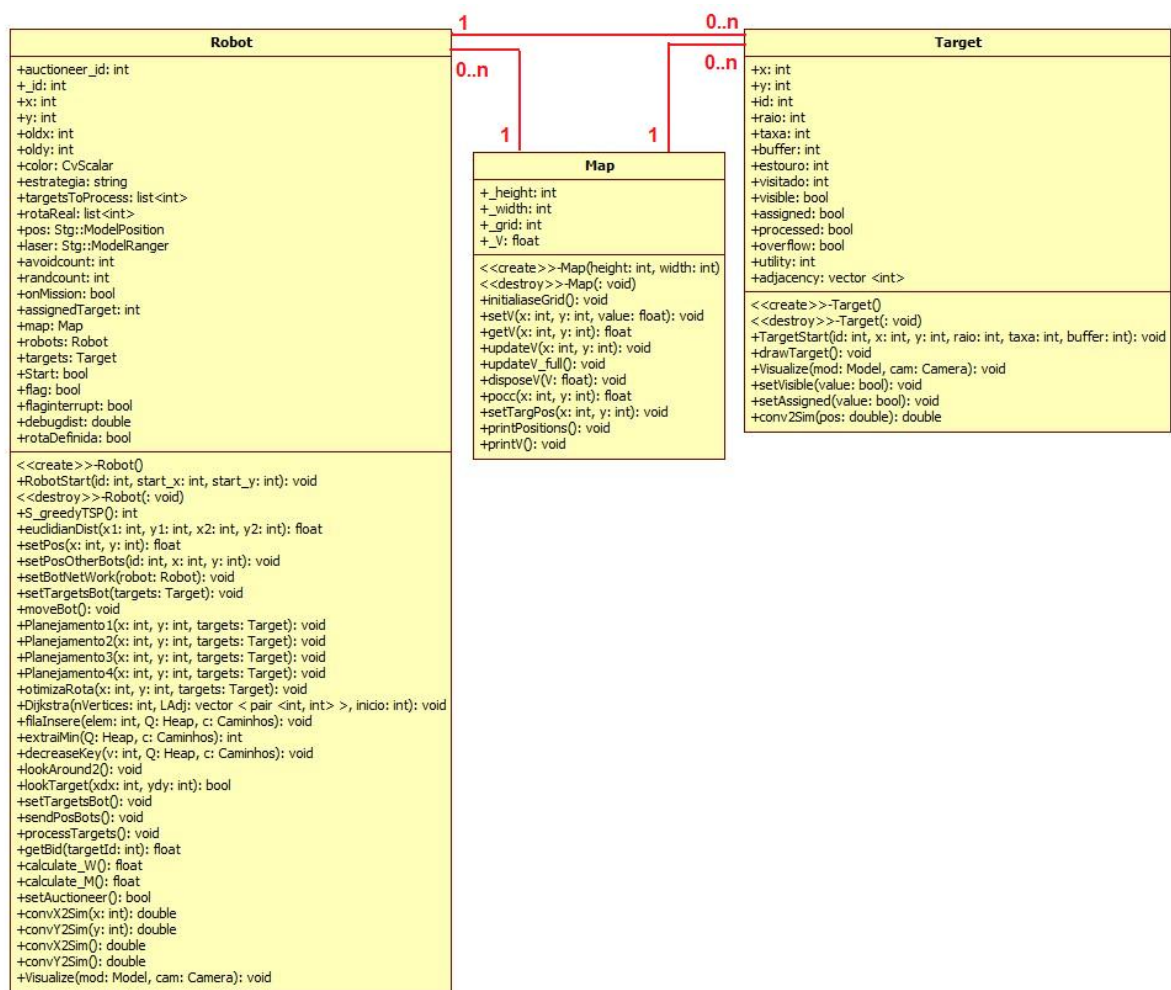


Figura 4.3: Diagrama de classes usadas no trabalho.

As simulações foram realizadas através das ferramentas Player e Stage. O Player fornece uma interface de rede para comunicação entre o código e o *hardware* dos robôs, muito utilizado

em robôs reais. Para simular uma população de robôs, é utilizado o Stage, que toma o lugar do *hardware* do robô real. Para se iniciar a simulação, alguns parâmetros são passados ao executar o Stage. Esses parâmetros informam ao código: o arquivo de entrada (contendo as informações de cada sensor), heurística escolhida e arquivo de saída. No arquivo de saída, serão gravados os dados da simulação: para cada sensor, é gravado o seu número identificador, momento em que foi visitado, e caso tenha ocorrido o estouro do *buffer* de memória, momento desse estouro e quantidade de dados perdidos até o momento da visita do robô móvel.

Passados esses parâmetros, o ambiente de simulação é inicializado. Primeiramente, o mapa onde os elementos estarão presentes. Em seguida, os demais elementos (robô e sensores) são inseridos no mapa. O robô pode ser colocado em uma posição aleatória, ou sua posição pode ser pré-determinada. Ao inserir os sensores, o programa lê do arquivo de entrada as respectivas posições e demais atributos. Um sensor pode estar em 4 estados diferentes:

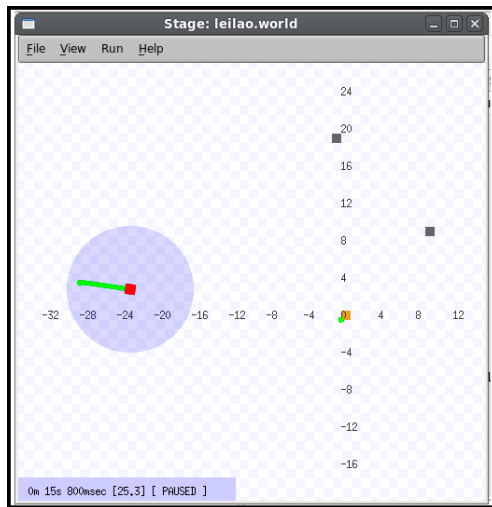
- *visible*: estado padrão inicial para todos os sensores.
- *assigned*: significa que esse sensor será o próximo a receber a visita do robô.
- *processed*: já foi visitado pelo robô e teve seus dados coletados.
- *overflow*: houve estouro na capacidade de armazenamento do *buffer* do sensor.

Em cada um desses estados, é atribuída uma cor diferente ao sensor. Dessa forma é possível acompanhar visualmente o andamento da coleta de dados, permitindo uma observação (mesmo que superficial) da simulação.

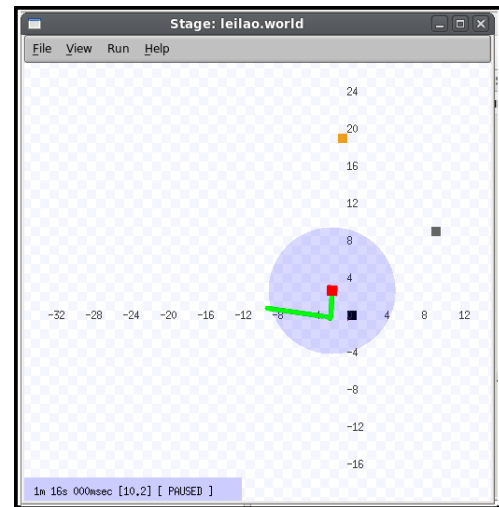
4.3 Cenários

O raio de comunicação r_i dos nós-sensores é fixo em todos os experimentos, igual a 10% da extensão do ambiente de simulação. O número de nós-sensores $n = i^2$ variou com $i = 3..10$, lembrando que quanto maior n maior a economia de distância percorrida quando se utiliza o Caixeiro Viajante com Vizinhos. O último parâmetro que foi variado foi a distribuição espacial dos nós-sensores. Para cada valor de n , foram geradas 26 distribuições espaciais distintas (uma regular e 25 aleatórias). Para cada configuração espacial, são gerados quatro cenários relativos a carga: armazenamento alto e carga homogênea (denominada Carga 1); armazenamento alto e carga heterogênea (Carga 2); armazenamento baixo e carga homogênea (Carga 3); armazenamento baixo e carga heterogênea (Carga 4). As métricas coletadas foram a distância percorrida e o total de perda de dados.

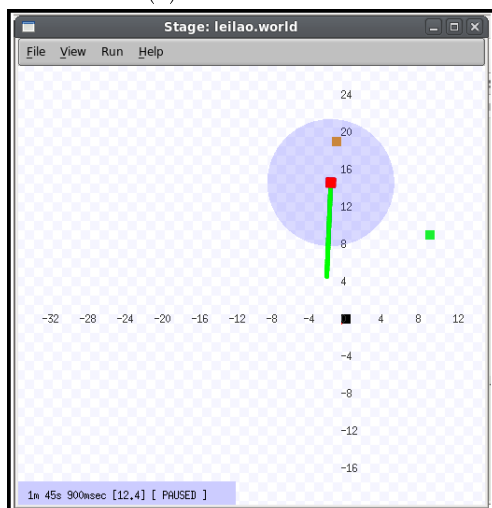
Cada nó sensor possui a sua própria frequência de aquisição, adquirindo os dados do ambiente a uma taxa, que pode ser diferente daquelas dos outros nós sensores presentes na rede. Para efeitos de simulação, foram considerados modelos de dois tipos de aquisição de dados: homogênea e heterogênea. A taxa de aquisição de dados homogênea é aquela que



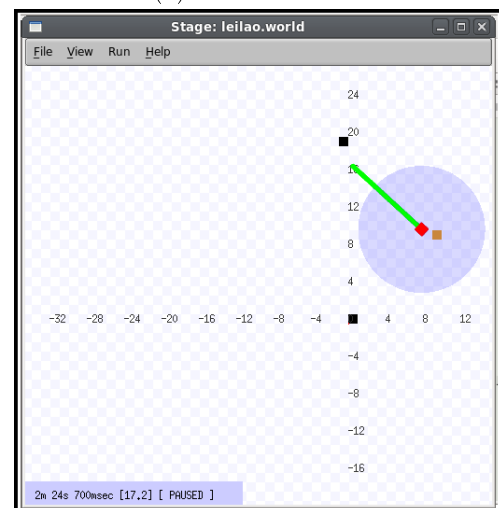
(a) 0m 15s 800msec



(b) 1m 16s 000msec



(c) 1m 45s 900msec



(d) 1m 24s 700msec

Figura 4.4: *Screenshots* capturadas durante uma simulação. O ponto vermelho indica o robô, enquanto a cor dos sensores varia de acordo com o estado em que se encontram. O tracejado verde junto ao robô indica um pedaço do caminho já percorrido pelo mesmo. A Figura 4.4a mostra o início da simulação, onde 2 sensores estão no estado *visible* (cinza) e o primeiro sensor a ter seus dados coletados se encontra no estado *assigned* (amarelo). Na Figura 4.4b, o primeiro sensor teve seus dados coletados e muda de estado, agora *processed* (preto), e o próximo a ser visitado também muda de estado. Enquanto o robô não alcança o segundo sensor, o terceiro tem seu *buffer* de memória esgotado, mudando para o estado *overflow*, indicado pela cor verde, como mostrado na Figura 4.4c. O momento do estouro é armazenado para que se possa calcular o quanto de dados foi perdido até o momento em que o robô alcançar esse sensor, mostrado na Figura 4.4d.

não sofre variação, ou seja, o fenômeno estudado não necessita que o nó sensor opere em diferentes frequências (fenômenos difusos). Um exemplo é um nó sensor que mede a umidade de um ambiente: ele vai adquirir os dados sempre na mesma taxa. A taxa de aquisição de dados heterogênea é aquela em que a aplicação ou fenômeno monitorado exige que o nó sensor opere em diferentes frequências, isto é, em um dado momento deve coletar mais dados que anteriormente (fenômenos concentrados). Um exemplo que exige essa taxa de aquisição de dados heterogênea é uma rede que monitora a temperatura para controle de incêndios em florestas: o nó sensor possui inicialmente uma taxa de aquisição de dados; a partir de uma certa temperatura, ele passaria a operar de forma mais “alerta” com uma taxa maior que a inicial, pois um aumento da temperatura pode indicar um foco de incêndio. Esses dois tipos de aquisição de dados serão considerados durante os testes das heurísticas de roteamento.

Nos experimentos em que os nós sensores tem a taxa de aquisição de dados homogênea, essa taxa terá um determinado valor igual para todos os nós. Para os experimentos em que os nós sensores tem a taxa de aquisição heterogênea, o *grid* da simulação é dividido em quatro quadrantes, sendo um deles sorteado aleatoriamente. Todos os nós sensores que estiverem localizados nesse quadrante sorteado, terão a taxa de aquisição de dados igual ao dobro dos demais, significando que essa área merece uma atenção maior (como por exemplo, uma área com grande risco de incêndios).

A distribuição espacial dos nós sensores serão de duas formas: uma aleatória e uma regular. Para a forma aleatória, todas as posições serão sorteadas; já para a segunda, a distribuição busca cobrir a maior área possível do ambiente da simulação. Para se chegar a essa distribuição regular, os nós sensores devem ocupar uma matriz de \sqrt{n} por \sqrt{n} nós, dada uma quantidade n de nós sensores.

Para cada uma das quantidades de nós sensores, todos terão o mesmo valor para o *buffer* de memória. Esse valor é adquirido em uma etapa inicial, antes das simulações, da seguinte maneira: os nós sensores são distribuídos de forma regular no ambiente; em seguida, o robô realiza a coleta dos dados utilizando o planejamento guloso do Caixeiro Viajante, sem considerar as taxas de dados ou vizinhança entre os nós; o tempo total k da coleta informa o valor do buffer. Esse valor pode ser utilizado de duas maneiras: em simulações em que os nós sensores tem grande quantidade de memória (onde a princípio nenhum deles sofreria perda de dados), todos tem esse valor igual a k ; em outras simulações, os nós sensores podem ter uma quantidade menor de memória, igual a 80% de k .

Em todos os experimentos, os nós sensores possuem o mesmo valor para o raio de comunicação, igual a 10% da extensão do *grid* de simulação. Como esse *grid* vai do valor -50 a 50 (tanto no eixo x quanto no y), o valor do raio de comunicação é estabelecido em 10 unidades para todos os nós sensores, em todas as simulações.

Capítulo 5

Experimentos e Resultados

Para a realização dos experimentos desse trabalho, diferentes cenários de teste foram abordados.

Dadas as simplificações deste trabalho, os parâmetros que afetam a coleta estão ligados aos nós sensores:

- quantidade, que influencia a complexidade de cálculo das heurísticas, mas por outro lado garantem uma maior cobertura do espaço e economia da heurística de vizinhança. Os experimentos são constituídos por 9, 16, 25, 36, 49, 64, 81 e 100 nós sensores;
- distribuição espacial, que influencia a qualidade da rota e a possibilidade de economia das heurísticas 3 e 4;
- taxas de aquisição de dados, que modelam o fenômeno observado, que pode ser difuso, com taxas constantes em todos os nós sensores, ou concentrado, onde eventos localizados aumentam a taxa de alguns dos nós sensores, dependendo de sua posição no espaço;
- capacidade de armazenamento dos nós sensores.

Para cada número de nós sensores, foram geradas 26 distribuições espaciais distintas (uma regular e 25 aleatórias). Para cada configuração espacial, são gerados quatro cenários relativos a carga: taxa de aquisição de dados, podendo ser homogênea (mesmo valor para todos) ou heterogênea (os nós sensores de um quadrante escolhido aleatoriamente terão o dobro dos demais); e tamanho do *buffer* de memória, podendo ser um valor alto (tempo total k definido anteriormente) ou baixo (80% de k). Logo, serão quatro tipos de carga: valor do *buffer* alto e taxa de aquisição de dados homogênea (denominada Carga 1); valor do *buffer* alto e taxa de aquisição de dados heterogênea (Carga 2); valor do *buffer* baixo e taxa de aquisição de dados homogênea (Carga 3); valor do *buffer* baixo e taxa de aquisição de dados heterogênea (Carga 4). As métricas coletadas foram a distância percorrida e o total de perda de dados. Dessa maneira, tem-se 8 quantidades de nós sensores, cada uma com 26 distribuições espaciais

distintas, com quatro tipos de carga. Logo, são $8*26*4 = 832$ cenários de teste distintos. Como foram realizadas 4 simulações para cada cenário de teste, tem-se ao total $4*832 = 3328$ simulações.

5.1 Resultados

Tabela 5.1: Tabela comparativa de heurísticas.

Carga	Carga 1		Carga 2		Carga 3		Carga 4	
Estratégias	Tempo	Perda	Tempo	Perda	Tempo	Perda	Tempo	Perda
1x2	100.00	100.00	109.99	25.38	100.00	100.00	109.99	50.12
1x3	89.97	108.82	89.97	73.64	89.97	55.77	89.97	73.34
1x4	89.97	108.82	99.92	22.35	89.97	55.77	99.92	36.49
2x4	89.97	108.82	90.72	94.50	89.97	55.77	90.72	67.48
3x4	100.00	100.00	110.91	27.84	100.00	100.00	110.91	46.36

A Tabela 5.1 compara a melhoria alcançada entre algumas heurísticas. As comparações foram feitas com as médias de tempo e perda de dados para os quatro tipos de carga: carga 1 (onde todos os nós têm a mesma taxa de aquisição de dados e valor de *buffer* alto), carga 2 (taxa de aquisição de dados dobrada para um dos quatro quadrantes sorteado aleatoriamente e valor de *buffer* alto), carga 3 (todos os nós possuem a mesma taxa de aquisição de dados e valor de *buffer* baixo) e carga 4 (taxa de aquisição de dados dobrada para um dos quatro quadrantes e valor de *buffer* baixo). Por exemplo: para comparar duas heurísticas i e j , primeiramente é calculada a média $m = \frac{t_{ik}}{t_{jk}}$ para cada quantidade k de nós sensores, onde: $k = 9, 16, 25, 36, 49, 64, 81$ e 100 ; e t_{ik} e t_{jk} representam as médias de tempo gasto na coleta utilizando as heurísticas i e j respectivamente com k nós. O valor de $m*100$ indica qual porcentagem do tempo gasto pela heurística j que a heurística i alcançou. Obviamente, se esse valor for maior que 100%, indica que i obteve um tempo maior que j . O mesmo cálculo foi realizado para a perda de dados.

Pode-se observar que a heurística 1 e a heurística 2 não apresentam diferenças quando os nós sensores estão monitorando fenômenos difusos, ou seja, todos os nós possuem a mesma taxa de aquisição de dados (tipos de carga 1 e 3). Como todos os nós sensores têm a mesma janela de tempo até que o *buffer* de memória sofra estouro, o planejamento acaba sendo o mesmo. Porém, para fenômenos concentrados, onde alguns nós sensores produzem mais dados que os demais (tipos de carga 2 e 4), a diferença é bem notável: a heurística 2 apresentou uma média de tempo levemente superior (pois o robô deve se deslocar primeiramente aos nós que produzem mais dados, podendo estes estarem muito longe da posição inicial do robô), mas uma diminuição significativa da perda de dados comparada à heurística 1.

Na comparação entre as heurísticas 1 e 3, a heurística 3 apresentou uma média de tempo

menor que o tempo gasto pela heurística 1 para os 4 tipos de carga. Essa diminuição de tempo foi a mesma em todos os tipos de carga, pois mesmo com alguns nós sensores produzindo mais dados que outros, apenas as vizinhanças eram consideradas. Comparando-se a perda, a heurística 3 apresentou um pequeno aumento para a carga 1, mas apresentou reduções nas demais.

Na comparação entre as heurísticas 1 e 4, a heurística 4 apresentou médias de tempo menores que a heurística 1, tanto para fenômenos difusos quanto concentrados. A heurística 4 apresentou uma perda de dados um pouco maior que a heurística 1 para a carga 1, mas apresentou melhorias satisfatórias nos demais resultados.

Em comparação com a heurística 2, a heurística 4 apresentou leve diminuição do tempo gasto pela heurística 2, tanto para fenômenos difusos quanto concentrados, havendo pouca diferença entre os resultados. A heurística 4 apresentou também uma perda um pouco maior nos cenários com carga 1, mas bons resultados nos demais cenários.

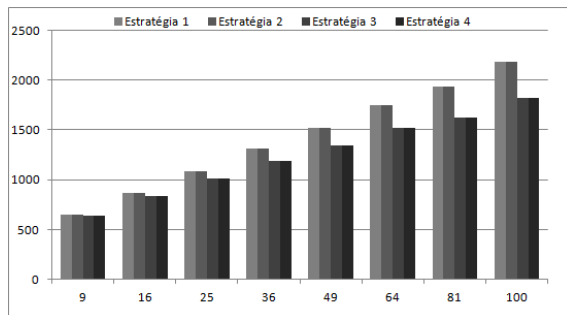
Finalmente, foram comparadas as heurísticas 3 e 4. As duas heurísticas não apresentaram melhorias uma sobre a outra nos cenários com tipo de carga 1 e 3, tanto em tempo quanto perda de dados. Já para os cenários com tipo de carga 2 e 4, a heurística 4 apresentou uma média de tempo maior que a heurística 3, mas com perdas de dados menores, reduzindo-a quase pela metade em um dos casos.

A Figura 5.1 apresenta as médias de tempo e perda de dados para cada tipo de carga. Pôde-se observar que, ao considerar a taxa de aquisição de dados de cada nó sensor, a perda de dados sofre uma redução muito grande em relação à perda sofrida quando usada a heurística que não considera esse fator. Além disso, as heurísticas que utilizam o conceito de vizinhança (que otimiza a rota do robô) têm uma melhoria ainda maior.

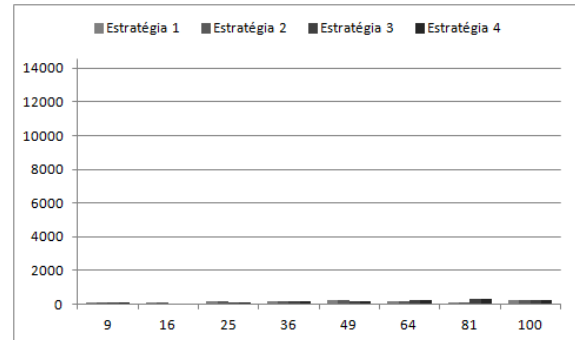
Para os tipos de carga 1 e 3, as heurísticas 1 e 2 obtiveram o mesmo resultado, assim como as heurísticas 3 e 4. Isso pode ser explicado pelo fato de que, como todos os nós possuem a mesma taxa de aquisição de dados, o planejamento que considera a carga vai se comportar como o planejamento guloso, pois sempre ocorre empate entre os nós sensores a sofrerem estouro de memória, escolhendo-se sempre o mais próximo. Ainda assim, o fato de considerar a vizinhança entre os nós (heurísticas 3 e 4) melhorou os resultados, tanto em tempo gasto para a realização da coleta quanto em perda de dados. Já para os tipos de carga 2 e 4, a heurística 2 teve um pequeno aumento de tempo em relação à heurística 1, o que ocorreu também com a heurística 4 em relação à 3. Isso ocorre porque nesses cenários, como alguns nós possuem taxa de aquisição de dados em dobro que os demais, o robô deve visitá-los primeiro, levando um tempo maior para alcançá-los. Mesmo com esse pequeno aumento de tempo, houve uma redução muito grande na perda de dados quando se compara a heurística 2 em relação à 1, assim como a 4 em relação à 3.

Com isso, é possível concluir que considerar a taxa de aquisição de dados de cada nó sensor produz uma diminuição considerável nas quantidades de dados perdidos em relação às heurís-

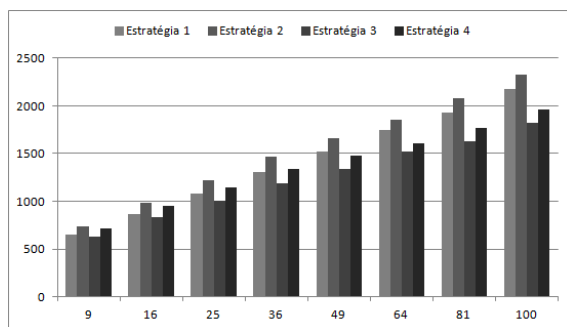
ticas que não utilizam esse parâmetro na definição da rota, mesmo que isso acarrete em um pequeno aumento do tempo gasto pelo robô para completar a coleta. Além disso, considerar a vizinhança dos nós pode melhorar ainda mais esses resultados, diminuindo (quando possível) o número de visitas e, conseqüentemente, o tempo e a distância percorrida pelo robô.



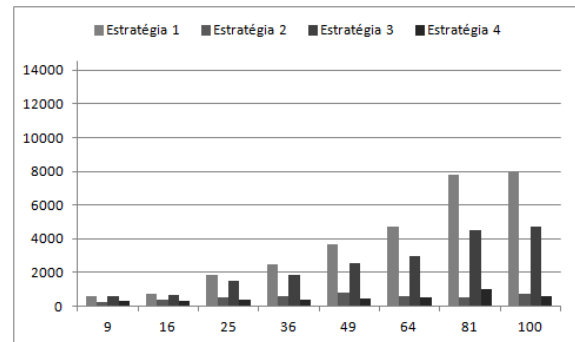
(a) Média de tempo gasto nos cenários com carga 1.



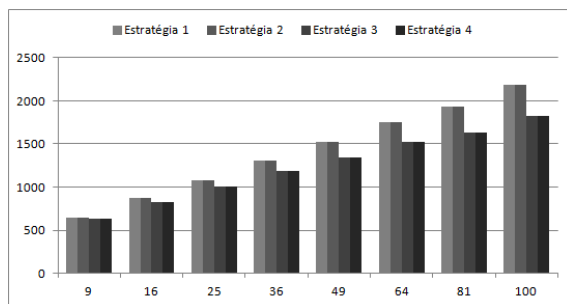
(b) Média de perda nos cenários com carga 1.



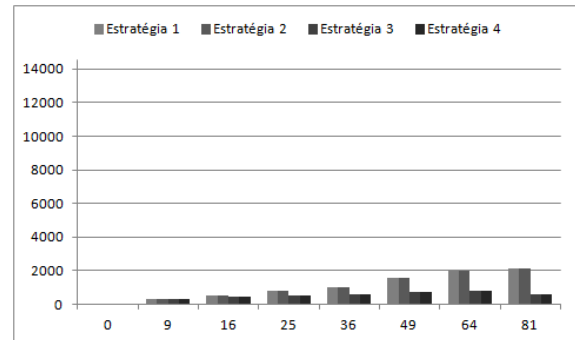
(c) Média de tempo gasto nos cenários com carga 2.



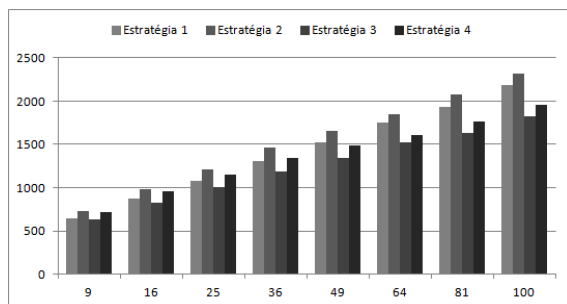
(d) Média de perda nos cenários com carga 2.



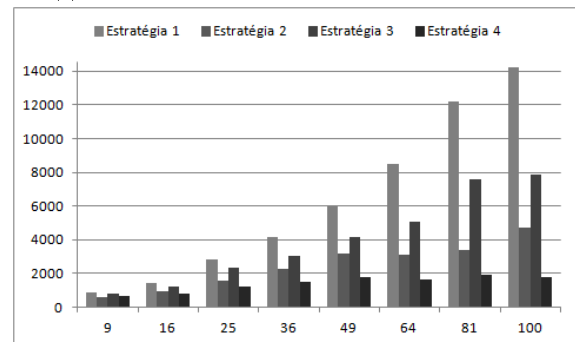
(e) Média de tempo gasto nos cenários com carga 3.



(f) Média de perda nos cenários com carga 3.



(g) Média de tempo gasto nos cenários com carga 4.



(h) Média de perda nos cenários com carga 4.

Figura 5.1: Gráficos apresentando as médias de tempo e perda de dados para cada um dos quatro tipos de carga.

Capítulo 6

Conclusões

O problema de estabelecer uma rota de visitas a nós sensores é um importante componente do problema de coleta de dados em redes de sensores sem fio por robôs móveis. Quando fenômenos orientados a eventos estão sendo monitorados, a taxa de aquisição de dados é diferente dependendo da posição espacial do nó sensor e neste trabalho, mostramos que levar em conta essa diferença pode produzir significativas melhoras no volume de dados coletados.

Para isso foi implementada uma heurística que integra tanto o raciocínio sobre nós sensores próximos da exaustão de memória, quanto define rotas otimizadas para visita de múltiplos nós sensores, diminuindo tanto a distância percorrida quanto a latência de entrega dos dados. Como resultado, mostramos que é possível reduzir significativamente a perda de dados sem aumentar a distância percorrida quando comparada à uma solução considerando a visita a todos os nós. A simplicidade da heurística permite que seja executada rapidamente, o que é altamente desejável para ambientes robóticos onde novas informações sobre o ambiente podem levar à necessidade de recálculo da rota.

A realização deste trabalho, mesmo com suas simplificações, foi importante devido às suas contribuições. Primeiramente, o modelo usado já está desenvolvido, podendo ser usado posteriormente em trabalhos que venham a estender este. Além do modelo, as implementações podem ser utilizadas em outros trabalhos, estendendo ou implementando novas heurísticas de roteamento. Juntamente com as implementações, a base de dados (cenários de teste utilizados e resultados obtidos) também podem ser utilizados para comparações com outras heurísticas utilizadas. Importante ressaltar que o trabalho também resultou em um artigo [9] submetido e publicado no IX Encontro Nacional de Inteligência Artificial (ENIA 2012), um dos eventos do Brazilian Conference on Intelligent Systems (BRACIS 2012). O artigo, intitulado “Estudo do impacto da carga no planejamento de rota para coleta de dados por robôs móveis em redes de sensores sem fio” está disponível no Apêndice.

Por ignorar as fontes de incerteza do problema, como conhecimento parcial do ambiente, erros de observação e atuação, posição desconhecida ou incerta dos nós sensores, a heurística aqui estabelecida pode não ser diretamente aplicável em ambientes reais.

Com a intenção de se estender o trabalho, as simplificações adotadas devem ser removidas em trabalhos futuros, com o intuito de se aproximar os resultados cada vez mais de um ambiente real. Novas abordagens de coleta de dados podem ser usadas, como por exemplo, uma abordagem que combina a coleta multi-salto com a coleta utilizando os robôs. Pretende-se também incluir um planejamento hierárquico: em áreas que vários nós sensores estão muito próximos uns dos outros, os dados amostrados são bem semelhantes. Assim, pode-se escolher qual dos nós deverá ter seus dados coletados pelo robô a cada rodada da coleta. Outro aspecto importante que deve ser integrado ao trabalho diz respeito à incerteza da localização, tanto dos nós sensores quanto do próprio robô ao longo da coleta. Nesse caso, o próprio robô deverá calcular sua posição e detectar a presença dos nós sensores, podendo ocorrer mudanças em suas posições.

Outra linha interessante de investigação, dado o desempenho bastante satisfatório mesmo de uma heurística simples, é investigar soluções aproximadas mais elaboradas existentes na comunidade de Otimização Exata e Meta-heurísticas para as formulações do problema do Caixeiro Viajante adotadas.

Apêndice

Artigo publicado nos anais do Brazilian Conference on Intelligent Systems (BRACIS 2012) e apresentado no IX Encontro Nacional de Inteligência Artificial (ENIA 2012), um dos eventos do BRACIS 2012.

Estudo do impacto da carga no planejamento de rota para coleta de dados por robôs móveis em redes de sensores sem fio

Gabriel Angelo Nazário¹, Andréa Iabrudi¹, Silmara Ribeiro¹

¹Departamento de Computação
Universidade Federal de Ouro Preto

andrea.iabrudi@iceb.ufop , {gabrielnazario25, silmara.ribeiro}@gmail.com

Abstract. *Good route planning is essential for data collection in wireless sensor networks by mobile robots to avoid loss of data and to minimize covered distance. The route must guarantee data to be collected from every sensor node, which data production rate depends on the observed phenomenon. In this work, we quantitatively evaluate if taking data load in account impacts in route quality. We explore different scenarios of spatial placement, storage capacity and homogeneity of load. The problem is formulated considering that communication can be established as soon as the robot enters node's communication range, making possible significant improvements in amount of collected data with the same covered distance.*

Resumo. *A coleta de dados por robôs móveis em rede de sensores sem fio exige planejamento de rotas, para evitar perda de dados e minimizar a energia consumida. A rota deve garantir a coleta dos dados em todos os nós sensores, cuja taxa de produção depende do fenômeno observado. Este trabalho avaliou quantitativamente o impacto de se considerar a taxa de produção de dados nas heurísticas para a definição da rota, em diferentes cenários de disposição espacial, capacidade de armazenamento e taxa de produção. Em conjunto com formulação do problema levando em conta o raio de comunicação dos nós sensores (sendo essa combinação totalmente nova), foi possível diminuir significativamente a perda de dados, sem impactar na distância percorrida.*

1. Introdução

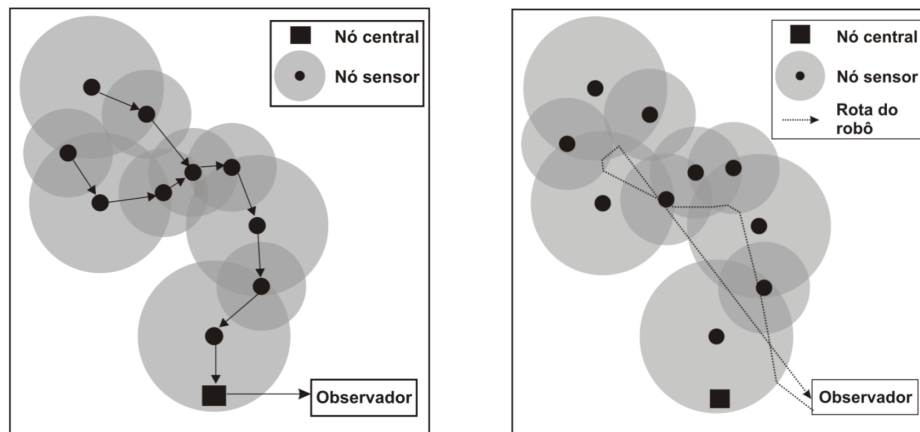
Neste trabalho mostramos que, numa rede de sensores sem fio (RSSF), considerar as características do fenômeno sob supervisão, quanto às taxas de produção de dados nos nós sensores, para definir a rota de coleta dos robôs móveis leva a diminuições consideráveis no volume de dados perdidos, sem impactar significativamente a distância total percorrida pelo robô. A introdução de mobilidade em uma RSSF é uma tendência recente, que permite a utilização de redes mais esparsas, já que não é necessário a garantia de conexão completa[Di Francesco et al. 2011]. Além de ser a única alternativa para acesso aos dados coletados em regiões desconectadas, a mobilidade é usada para aumentar a confiabilidade da rede, e ao mesmo tempo diminuir o custo (muito menos nós sensores) e o gasto de energia (não há necessidade de comunicação multi-salto). Esses benefícios só são alcançados se os desafios introduzidos pelos elementos móveis, como detecção de contato com sensores, transferência confiável de dados, forem resolvidos a contento.

Neste trabalho, nos concentramos na questão de controle de mobilidade quando um único robô móvel é responsável pela coleta de dados de um conjunto definido de nós sensores, ou seja, abordamos o problema do planejamento da rota a ser percorrida por esse robô a fim de coletar os dados com a menor perda e o menor consumo próprio de energia, levando em conta taxa de produção de dados, ou carga, de cada nó sensor presente na rede, aspecto pouco abordado na literatura.

As RSSF são cada vez mais utilizadas, pois são opções simples e baratas para monitoramento e sensoriamento de diversos fenômenos (umidade, poluição, tráfego) ou detecção de eventos e rastreamento (invasões, incêndios, movimentações de animais). Elas são compostas por um conjunto grande de nós sensores, pequenos dispositivos alimentados a bateria, que realizam medições, armazenamento e processamento local e comunicação de dados para um dispositivo específico chamado base. Cada sensor tem sua própria carga, volume de dados amostrados num intervalo de tempo, dependendo do fenômeno e de sua posição espacial. Em atividades de sensoriamento e monitoramento, a carga tende a ser a mesma em todos os nós sensores e a rede é dita homogênea. Quando a carga varia, a rede é heterogênea, o que acontece tipicamente em detecção de eventos e rastreamento. Após a medição local, os dados podem ser processados e imediatamente enviados para a base, ou armazenados para envio posterior. Cada sensor possui memória limitada e o tempo decorrido antes da perda de dados depende do instante da última coleta e da sua carga.

A forma mais comum de transmissão é a instantânea multi-salto (Figura 1a), onde os dados são enviados assim que são processados e trafegam pela rede passando de nó em nó, até atingir a base. Como o maior consumo de energia em RSSF envolve atividades de comunicação, essa alternativa gera um gasto elevado de energia, pois cada nó sensor deve receber os dados dos nós vizinhos e retransmiti-los. Além disso, os nós sensores próximos à base tendem a ser mais utilizados e podem levar à rápida desconexão da rede. Com a utilização de robôs móveis para coleta, os dados são armazenados localmente no nó-sensor até que o robô atinja a sua área de comunicação, como mostrado na Figura 1b. Os dados são então transmitidos ao robô que eventualmente os entrega à base. Para que haja eficiência na coleta – pouca perda de dados e pouca energia gasta pelo robô – um aspecto fundamental é a definição da ordem na qual os nós-sensores serão visitados, chamado de problema de planejamento de rota.

Esse problema vem sendo abordado sob diversas perspectivas diferentes [Di Francesco et al. 2011], visando aumentar tempo de vida da rede, diminuir latência de entrega de dados no sumidouro, aumentar a cobertura, evitar obstáculos, além dos aspectos de coleta colaborativa. As abordagens vão desde uma política quase reativa [Soares et al. 2007] até planejamento sofisticado levando em conta a latência [Sugihara and Gupta 2011]. Nesse trabalho enfatizamos principalmente minimização da perda de dados e as características do fenômeno monitorado que podem influenciar no roteamento. A taxa de produção de dados já foi tratada anteriormente em [Black et al. 2006, Somasundara et al. 2007], onde o problema foi modelado como um Caixeiro Viajante com Janela de Tempo [Solomon 1987]. Mais recentemente, várias abordagens utilizam o modelo de Caixeiro Viajante com Vizinhos para representar a possibilidade de se comunicar com vários nós sensores em um único ponto [Yuan et al. 2007, Sugihara and Gupta 2011]. Nossa abordagem para o problema



(a) Envio de dados dos sensores de forma multi-salto.

(b) Coleta de dados com ajuda de robôs.

Figura 1. Possíveis maneiras de se coletar dados em redes de sensores sem fio: através da abordagem multi-saltos (Figura 1a), onde os dados são passados de um nó sensor ao outro, até chegarem à base; utilização de um robô móvel, que percorre o ambiente coletando os dados dos nós sensores (Figura 1b).

do planejamento de rota integra as duas perspectivas, apresentando uma solução para o problema do caixeiro viajante com vizinhos e janela de tempos.

Foram definidas heurísticas para quatro versões do problema: a primeira considera apenas as distâncias entre os elementos; a segunda considera também carga e capacidade de armazenamento de cada nó-sensor; a terceira permite coleta de dados de todos os nós sensores cuja área de comunicação seja alcançável pelo robô, estabelecendo uma variação do caixeiro viajante com vizinhos; e finalmente a quarta versão e nova heurística proposta neste trabalho integra todos os aspectos no mesmo problema. As heurísticas foram implementadas e simuladas em diferentes cenários de teste, onde há variação de número de nós sensores e características do fenômeno observado.

Na próxima seção, definimos mais formalmente o problema abordado. Em seguida, são apresentadas as quatro heurísticas utilizadas para o planejamento de rotas do robô. Foram realizados experimentos utilizando simulação, com diferentes cenários, e os resultados são apresentados na Seção 4. O artigo termina com uma discussão dos resultados obtidos e sugestões de trabalhos futuros.

2. O problema de planejamento de rota de coleta

Uma RSSF com mobilidade envolve como elementos principais o ambiente, os nós-sensores, o fenômeno sob observação e o robô. Numa aplicação real, são inúmeras as fontes de incerteza – sensores do robô limitados e imprecisos, aspectos do ambiente desconhecido, posição exata dos nós-sensores desconhecida, fenômeno sob observação parcialmente caracterizado – e dinamismo – os nós-sensores podem se deslocar por influência externa, o ambiente pode se alterar. As decisões a serem tomadas são bastante variadas e complexas: quais nós-sensores devem ser visitados para garantir menor consumo de energia mantendo o mesmo nível de informação, adoção de mecanismos mistos de ro-

teamento, com multi-salto aliado a coleta por robô móvel, exploração do ambiente para definir localização de nós sensores. Estes aspectos bem como outras considerações quanto aos mecanismos de comunicação podem ser encontrados em [Di Francesco et al. 2011].

Neste trabalho, várias simplificações são feitas, com enfoque na carga de dados como variável de interesse no planejamento de rota. O modelo proposto é determinístico e estático, e o ambiente é conhecido, incluindo posição, capacidade de armazenamento e comunicação dos nós-sensores e carga de dados induzida pelo fenômeno. O robô é caracterizado pela sua velocidade, considerada constante, e sua posição no ambiente, conhecida com exatidão, ou seja, são ignorados os problemas de mapeamento e localização. O ambiente é sem obstáculos e discreto, representado por um *grid* de tamanho fixo, e cada elemento, robô ou nó-sensor, é localizado pela célula onde se encontra. O planejamento da rota do robô considerando as distâncias entre centróides de células.

Mais formalmente, uma RSSF possui n nós-sensores, onde cada nó i é caracterizado por sua posição fixa (x_i, y_i) , seu raio de comunicação r_i e seu limite de armazenamento b_i . O fenômeno sob supervisão é caracterizado pelo vetor n -dimensional c da carga de dados induzida na posição de cada nó-sensor c_i por unidade de tempo. A carga é considerada constante durante o período de coleta. O robô tem uma posição (x_r, y_r) e uma velocidade constante v e possui conhecimento sobre todos parâmetros do sistema descritos acima.

O planejamento de rota consiste em definir a ordem na qual os n de nós serão visitados para coleta de dados. A distância percorrida pelo robô ao implementar uma rota é considerada como a soma entre as distâncias entre os nós sucessivos mais a distância inicial do robô até o primeiro nó. Esse valor é um limite superior para a distância real percorrida, já que os dados de um nó-sensor i são coletados instantaneamente assim que o robô atinja sua área de comunicação, ou seja, quando a distância entre ambos seja menor ou igual a r_i , não sendo necessário atingir a posição exata do nó-sensor (veja exemplo na Figura 2a-b). O instante em que um nó i é visitado depende da rota, dos raios de comunicação e da velocidade do robô. Seja t_i o instante da visita, se o volume de dados induzido pelo fenômeno $c_i \times t_i$ for menor que o limite de armazenamento b_i , não houve perda de dados. Caso contrário, a perda é de $c_i \times t_i - b_i$.

Quando a única medida quantitativa da qualidade da rota de coleta é a distância percorrida pelo robô, ou seja, deseja-se apenas minimizar a energia consumida pelo robô sem preocupação com perda de dados, o problema é modelado como um Caixeiro Viajante [Lawler et al. 1985] clássico, onde a matriz de distâncias é obtida pela distância entre os nós-sensores. Há diversas soluções exatas, aproximadas e heurísticas para o problema que podem ser utilizadas. Quando além da distância, deseja-se minimizar a perda de dados, o problema se aproxima da formulação do Caixeiro Viajante com janelas de tempo [Solomon 1987], onde o nó i tem janela que varia entre 0 e $\lceil \frac{b_i}{c_i} \rceil$, para que seja evitada perda. Mais recentemente, incorporou-se à modelagem do problema a observação de que um nó-sensor pode ter seus dados coletados, mesmo que esse nó não seja o próximo da rota do robô, desde que este último tenha entrado em sua área de comunicação. Esse problema é um caso especial do Problema do Caixeiro Viajante com Vizinhanças tratado por [Sugihara and Gupta 2011].

Nossa proposta de solução integra os dois modelos, considerando simultanea-

mente a questão de perda de dados e de comunicação precoce, resolvendo uma variação que pode ser denominada de Problema do Caixeiro Viajante com Vizinhanças e Janela de Tempo, que visa minimizar a distância percorrida enquanto evita a perda de dados, dando assim prioridade de visita a nós com menor capacidade de armazenamento e/ou alta carga de dados.

3. Heurísticas de planejamento

O planejamento da rota é feito de forma *offline*, ou seja, antes de começar a coleta, em uma etapa inicial e o robô executa exatamente a ordem de visita prescrita pela heurística. Foram implementadas quatro soluções heurísticas, uma para cada variação de problema descrita acima. A última variação não havia sido tratada até o momento na literatura, até onde sabemos. É importante ressaltar que o maior interesse deste trabalho é avaliar em que situações considerar a carga e capacidade de armazenamento dos nós-sensores no planejamento resulta em uma menor perda de dados.

A matriz de distâncias em todas as heurísticas é obtida calculando-se diretamente a distância entre os nós-sensores e entre estes e a posição inicial do robô. Considera-se que o ponto de partida do robô é a base para onde se quer enviar os dados, logo ele deve retornar ao ponto de partida para terminar a coleta. Na versão Caixeiro Viajante clássico, que serve como limite superior para perda de dados, foi proposta a **Heurística 1**, solução gulosa bastante simples, onde o critério guloso é o nó-sensor que será alcançado no menor tempo, ou seja, aquele mais próximo da posição atual, já que a velocidade do robô é sempre constante.

A **Heurística 2** é também solução gulosa, mas para o Problema do Caixeiro Viajante com Janela de Tempo. Como critério guloso é utilizada a diferença entre o tempo de esgotamento da capacidade de armazenamento, definida como autonomia do nó-sensor, e o tempo para atingí-lo. Ou seja, se dois nós estão à mesma distância do robô, ele irá se dirigir para aquele com menor autonomia de armazenamento. Se ambos têm a mesma autonomia, ele irá se dirigir para o que está mais distante. O comportamento esperado da heurística é diminuir a perda de dados, acarretando contudo uma maior distância percorrida.

É importante ressaltar que para ambas heurísticas, a implementação efetiva da rota prescrita para o robô não terá a mesma distância avaliada no planejamento, já que não há necessidade de navegar até a posição exata do nó-sensor, bastando atingir sua área de comunicação. Por exemplo, na Figura 2(a) temos um cenário onde uma rota inicial foi escolhida, definindo a trajetória em tempo de execução mostrada na Figura 2(b).

O Problema do Caixeiro Viajante com Vizinhanças captura exatamente esta observação, de que não é necessário visitar todos os nós-sensores, pois é possível uma visita indireta aos vizinhos de um nó qualquer, desde que a respectiva área de comunicação seja atingida. Nossa terceira heurística parte de uma rota inicial qualquer para definir o grafo de vizinhança. Seja o j -ésimo nó-sensor da rota. Ele será vizinho de um nó subsequente, que ocupa a posição $i \geq j$, caso seja possível para o robô se comunicar com todos os nós do intervalo $[j + 1, i - 1]$ na sua trajetória de j para i , ou seja, se o segmento de reta que une a posição dos dois sensores intercepta as áreas circulares de comunicação dos nós intermediários. Se isso é verdade, é desnecessária a visita aos nós intermediários. Nesse caso, a aresta (i, j) é adicionada ao grafo, com o custo da aresta

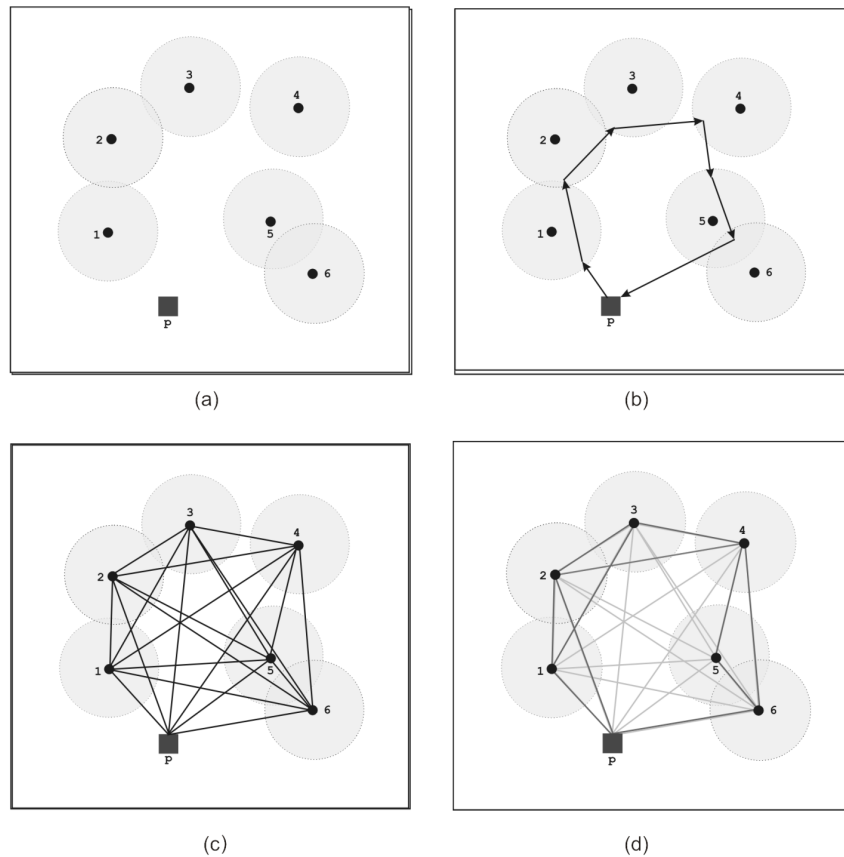


Figura 2. Exemplo genérico de otimização sobre uma rota gerada. Inicialmente, é gerada a rota de visitas por alguma das estratégias apresentadas anteriormente (Figura 2a e Figura 2b). Em seguida, é montado um grafo, adicionando arestas que possam eliminar algumas visitas, desde que o robô passe pelo raio de comunicação do nó sensor. Depois do grafo pronto, o algoritmo de menor caminho de Dijkstra é utilizado, para que o robô possa percorrer a rota otimizada.

sendo a distância entre esses dois nós-sensores. Montado o grafo, o algoritmo de Dijkstra é utilizado para se encontrar o menor caminho, que por construção, será menor ou igual à distância da rota inicial, mantendo a característica de coleta de dados de todos os nós-sensores, implementando assim a **Heurística 3**. O tipo de otimização realizado é ilustrado na Figura 2(c-d). Os resultados obtidos em[Sugihara and Gupta 2011] indicam redução significativa de distância total percorrida quando a abordagem é utilizada, resultados que reproduzimos nos nossos experimentos.

A principal contribuição deste trabalho, quanto ao planejamento de rotas, é integrar a otimização por vizinhanças a considerações sobre carga, obtendo-se um planejamento que minimiza simultaneamente a perda de dados e a distância percorrida na **Heurística 4**. A heurística proposta integra a solução gulosa para um problema de Caixeiro Viajante com Janela de Tempo com uma otimização posterior que elimina nós dessa rota, que podem ser atingidos ao visitar seus vizinhos. Resta verificar a eficiência de tal abordagem, principalmente com diferentes características de RSSF em relação à produção

de dados e capacidade de armazenamentos. Para isso foram realizados experimentos simulados, com avaliação das quatro heurísticas com diferentes cenários de carga.

4. Simulação e Experimentos

Na avaliação da heurística proposta, utilizou-se o ambiente de simulação Player/Stage, bastante popular na comunidade de robótica. As ferramentas Player e Stage (<http://playerstage.sourceforge.net/>) foram utilizadas para a realização das simulações. O Player fornece uma interface de rede para uma grande variedade de *hardware* de robôs e sensores reais. O modelo cliente/servidor permite que os programas de controle do robô possam ser escritos em várias linguagens de programação (entre elas Java, C, C++ e Python) e executados em qualquer computador com uma conexão de rede para o robô. O Stage simula uma população de robôs móveis que se deslocam em um ambiente bidimensional. Os dispositivos presentes no Stage apresentam uma interface padrão com o Player, o que permite que os programas de controle usados na simulação possam ser transferidos para o *hardware* de um robô real, necessitando de poucas ou nenhuma alteração em seu código. O ambiente de simulação é um quadrado de tamanho fixo para todos os experimentos e é abstratamente visualizado como um grid de $\sqrt{n} \times \sqrt{n}$ células.

De especial cuidado o estabelecimento de diferentes cenários para categorizar cargas induzidas por diferentes fenômenos, bem como o impacto do número e capacidade de armazenamento dos nós-sensores. Nas simulações foram consideradas quatro tipos de carga, combinando cargas homogêneas ou heterogêneas e capacidade de armazenamento alta ou baixa. Com isso, caracterizamos fenômenos difusos, que acontecem por toda a região de interesse com igual intensidade e onde os nós não diferem entre si quanto ao momento de estouro, ou concentrados, que geram mais carga em alguma sub-região cujos nós-sensores ali inseridos tipicamente deveriam ser visitados primeiro. Nos experimentos o primeiro caso foi caracterizado por uma carga de dados c_i igual para todos os sensores. No segundo caso, um dos quatro quadrantes da região de simulação foi aleatoriamente escolhida para indicar a concentração do fenômeno. A todos os nós-sensores ali posicionados foi atribuída o dobro da carga dos outros quadrantes.

Sob outra perspectiva, comparamos redes com grande capacidade de armazenamento versus aquelas onde há menor capacidade, onde heurísticas que minimizem a perda deveriam fornecer melhor resultado. Para grande capacidade, foi atribuído a cada nó-sensor capacidade de armazenamento suficiente para armazenar todo o dado gerado por um fenômeno difuso caso o robô percorresse todas as células do grid. Para caracterizar cenários de pouca capacidade de armazenamento, foi considerado 80% desse valor.

O raio de comunicação r_i dos nós-sensores é fixo em todos os experimentos, igual a 10% da extensão do ambiente de simulação. O número de nós-sensores $n = i^2$ variou com $i = 3..10$, lembrando que quanto maior n maior a economia de distância percorrida quando se utiliza o Caixeiro Viajante com Vizinhos. O último parâmetro que foi variado foi a distribuição espacial dos nós-sensores. Para cada valor de n , foram geradas 26 distribuições espaciais distintas (uma regular e 25 aleatórias). Para cada configuração espacial, são gerados quatro cenários relativos a carga: armazenamento alto e carga homogênea (denominada Carga 1); armazenamento alto e carga heterogênea (Carga 2); armazenamento baixo e carga homogênea (Carga 3); armazenamento baixo e carga hete-

rogênea (Carga 4). As métricas coletadas foram a distância percorrida e o total de perda de dados.

4.1. Discussão

A Figura 3, as médias de tempo(distância) de rota e perda de dados para 26 instâncias de diferentes configurações espaciais para cada tipo de carga e número de sensores são apresentadas e cada tipo de heurística (ou estratégia). As estratégias que levam em conta a carga são a 2 e 4, enquanto as estratégias 3 e 4 aproveitam-se da vizinhança. Pode-se observar uma significativa redução da perda de dados quando essas heurísticas são aplicadas a cenários onde ou o fenômeno é concentrado ou a capacidade de armazenamento é baixa. O tempo de rota da Heurística 2 é superior ao da que não considera carga, contudo, nossa heurística que combina também a vizinhança (Heurística 4), diminui esse tempo além de diminuir a perda de dados, atingindo simultaneamente os dois objetivos. A utilização de vizinhança diminui consistentemente o tempo de rota, assim como a consideração do armazenamento diminui consistentemente a perda de dados.

Para fenômenos difusos (Carga 1 e 3), as Heurísticas 1 e 2 obtiveram resultados de tempo e perda equivalentes, assim como as Heurísticas 3 e 4. Isso pode ser explicado pelo fato de que, como todos os nós possuem a mesma taxa de produção de dados, o planejamento que considera a carga vai se comportar como o planejamento guloso, pois sempre ocorre empate entre os nós sensores a sofrerem estouro de memória, escolhendo-se sempre o mais próximo. Ainda assim, o fato de considerar a vizinhança entre os nós (Heurísticas 3 e 4) tem melhor desempenho nos dois quesitos. Os tempos são equivalentes para Carga 1 e 3, e a perda de dados é, como esperado, maior no segundo caso, mas proporcional ao número de sensores, o que indica que a capacidade local de armazenamento não tem impacto sobre o desempenho.

Já para fenômenos concentrados (Carga 2 e 4), o tempo de rota foi ligeiramente maior para as heurísticas que consideram a carga, quando comparadas diretamente àquelas que consideram somente distância – Heurística 2(4) em relação a Heurística 1(3). Isso ocorre porque nesses cenários, como alguns nós possuem taxa de produção de dados em dobro que os demais, o robô deve visitá-los primeiro, levando um tempo maior para alcançá-los. Mesmo com esse pequeno aumento de tempo, houve uma redução muito grande na perda de dados quando se compara a estratégia 2 em relação à 1, assim como a 4 em relação à 3, indicando que para esses cenários é muito importante levar em conta a carga no planejamento de rota. Nessas situações há uma diferença mais marcada na perda de dados quando utilizam-se heurísticas que não levam em conta carga para armazenamento alto e baixo, mas a utilização da Heurística 4 consegue quase equiparar a perda.

Com isso, é possível concluir que considerar a carga de dados de cada nó-sensor, em conjunto com a vizinhança para introdução de atalhos na rota, produz uma diminuição considerável da perda de dados, sem contudo implicar em aumento significativo do tempo gasto para a coleta. A diminuição da perda é maior para fenômenos concentrados e com capacidade mais baixa de armazenamento.

A Tabela 1 apresenta um sumário dos resultados, comparando duplas de heurísticas em cada linha. A linha 1 \times 2 apresenta os resultados relativos da Heurística 2 em relação à Heurística 1, ressaltando o impacto da introdução da consideração da carga,

Tabela 1. Tabela comparativa de heurísticas.

Carga	Carga 1		Carga 2		Carga 3		Carga 4	
Estratégias	Tempo	Perda	Tempo	Perda	Tempo	Perda	Tempo	Perda
1x2	100.00	100.00	109.99	25.38	100.00	100.00	109.99	50.12
1x3	89.97	108.82	89.97	73.64	89.97	55.77	89.97	73.34
1x4	89.97	108.82	99.92	22.35	89.97	55.77	99.92	36.49
2x3	89.97	108.82	81.75	412.60	89.97	55.77	81.75	153.89
2x4	89.97	108.82	90.72	94.50	89.97	55.77	90.72	67.48
3x4	100.00	100.00	110.91	27.84	100.00	100.00	110.91	46.36

mesmo efeito ressaltado na linha 3×4 , mas agora sobre uma Heurística que já utiliza vizinhança. Já as linhas 1×3 e 2×4 evidenciam o efeito da introdução da vizinhança. Já a linha 2×3 mostra a comparação entre somente vizinhança e somente carga, enquanto a linha 1×4 mostra a melhoria final de nossa abordagem em comparação à abordagem mais simples. Todos os resultados são a razão entre os valores médios das duas heurísticas multiplicadas por 100 (média entre 26 instâncias). Para o tempo, valores menores do que 100 indicam que a segunda heurística é superior. Para a carga, valores menores do que 100 indicam que a segunda heurística é inferior.

Ao analisar separadamente a introdução de uma política de consideração de carga, (Heurística 2 em relação à Heurística 1 e Heurística 4 em relação à Heurística 3), não há diferenças quando todos os sensores possuem a mesma carga de dados - fenômenos difusos (tipos de carga 1 e 3), pois todos têm a mesma janela de tempo até que o estouro. Porém, para fenômenos concentrados (tipos de carga 2 e 4), a diferença é bem notável: com um incremento de tempo de 10%, houve uma significativa redução da perda de dados: cerca de 25% da perda inicial no cenário onde a capacidade de armazenamento é alta (Carga 2) e cerca de 50% para capacidade de armazenamento reduzida. É interessante notar que o impacto percentual é o mesmo, caso seja utilizada a otimização por vizinhos, o que é explicado pela dependência forte da qualidade da rota de entrada no resultado final.

A introdução de economia de navegação devido à utilização de comunicação com vizinhos (Heurística 3 em relação à Heurística 1 e Heurística 4 em relação à Heurística 2), notamos que o comportamento é bastante parecido para fenômenos difusos: uma redução de cerca de 10% no tempo total da rota em todos os casos, uma diminuição significativa da perda com capacidade reduzida, porém com um pequeno (e surpreendente) aumento da perda quando a capacidade de armazenamento é alto. Para explicar este resultado, as perdas nesse caso são muitíssimo pequenas e a diferença entre as médias não é estatisticamente significativa. Já para os fenômenos concentrados, o comportamento continua semelhante em relação ao tempo (10% de melhora), mas o impacto da escassez de capacidade armazenamento tem mais impacto quando a carga é levada em conta no planejamento (Heurística 4 em relação à Heurística 2), quando há uma redução de cerca de 35% na perda.

A comparação direta entre as Heurísticas 2 e 3 mostra que há uma consistente redução do tempo de rota - o que é esperado. Há também um aumento da perda de dados, principalmente para Carga 2, onde o efeito de planejamento é mais impactante, levando a uma perda 400% maior. A exceção é a diminuição com Carga 3, que pode ser explicado

por uma melhora da perda devido à diminuição da rota.

Ao considerar nossa heurística em comparação àquela onde nem carga nem vizinhança foram levadas em consideração (1×4), houve uma redução bastante significativa redução da perda de dados (os resultados para Carga 1 não tem significância estatística, de até 80%, enquanto não houve aumento no tempo de rota em nenhuma das cargas. Os dados experimentais corroboram a nossa hipótese de que levar em conta a carga no planejamento é bastante importante para um bom desempenho quanto a perda de dados, enquanto utilizar a vizinhança permite que o tempo total de rota não seja aumentado.

5. Conclusão

O problema de estabelecer uma rota de visita a nós sensores é importante componente do problema de coleta de dados em redes de sensores sem fio por robôs móveis. Quando fenômenos orientados a eventos estão sendo monitorados, a taxa de aquisição de dados é diferente dependendo da posição espacial do nó sensor e neste trabalho, mostramos que levar em conta essa diferença pode produzir significativas melhoras no volume de dados coletados.

Para isso foi implementada uma heurística que integra tanto o raciocínio sobre nós sensores próximos da exaustão de memória, quanto define rotas otimizadas para visita de múltiplos nós sensores, diminuindo tanto a distância percorrida quanto a latência de entrega dos dados. Como resultado, mostramos que é possível reduzir significativamente a perda de dados sem aumentar a distância percorrida quando comparada à uma solução considerando a visita a todos os nós. A simplicidade da heurística permite que seja executada rapidamente, o que é altamente desejável para ambientes robóticos onde novas informações sobre o ambiente podem levar a necessidade de recálculo da rota.

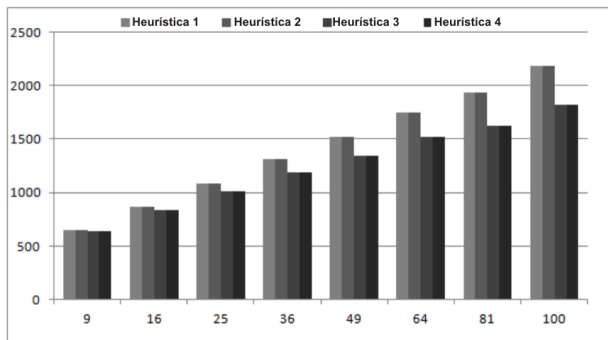
Por ignorar as fontes de incerteza do problema, como conhecimento parcial do ambiente, erros de observação e atuação, posição desconhecida ou incerta dos nós sensores, a heurística aqui estabelecida pode não ser diretamente aplicável e pretendemos estender o modelo para incorporar essas incertezas.

Outra linha interessante de investigação, dado o desempenho bastante satisfatório mesmo de uma heurística simples, é investigar soluções aproximadas mais elaboradas existentes na comunidade de Pesquisa Operacional e Meta-heurísticas para as formulações do problema do Caixeiro Viajante adotadas.

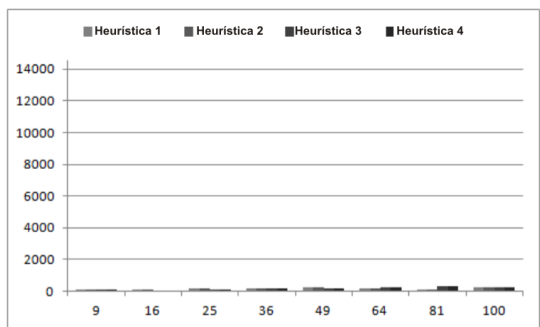
Referências

- Black, T., Mak, V., Pathirana, P. N., and Nahavandi, S. (2006). Using autonomous mobile agents for efficient data collection in sensor networks. *World Automation Congress (WAC), Budapest, Hungary*.
- Di Francesco, M., Das, S. K., and Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):7.
- Lawler, E., Lenstra, J. K., Kan, A. H. G. R., and Shmoys, D. B. (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.

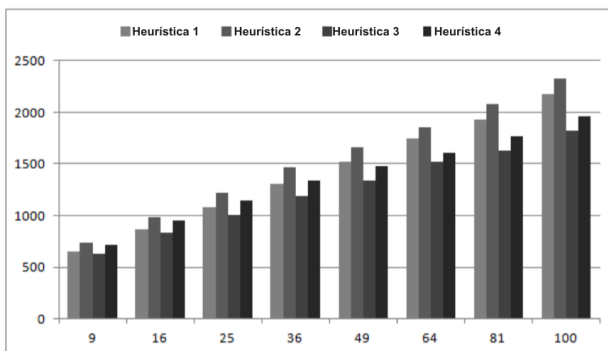
- Soares, M. B., Campos, M. F. M., Dutra, D. A., Campos, V. C. d. S., and Pereira, G. A. S. (2007). Hybrid mobile robot navigational strategy for efficient data collection in sparsely deployed sensor networks. In *IROS'07*, pages 2833–2838, San Diego, CA, USA.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, page 254–265.
- Somasundara, A., Ramammorthy, A., and Srivastava, M. B. (2007). Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410.
- Sugihara, R. and Gupta, R. K. (2011). Path planning of data mules in sensor networks. *ACM Transactions on Sensor Networks*, 8(1):1–27.
- Yuan, B., Orlowska, M., and Sadiq, S. (2007). On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*.



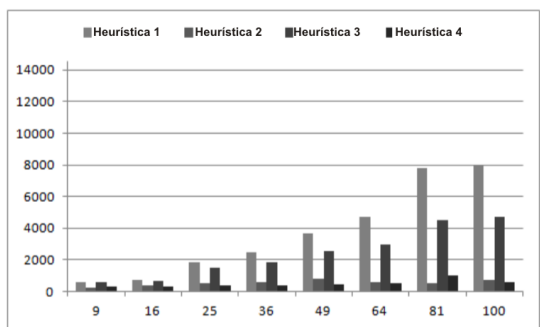
(a) Média de tempo gasto nos cenários com carga 1.



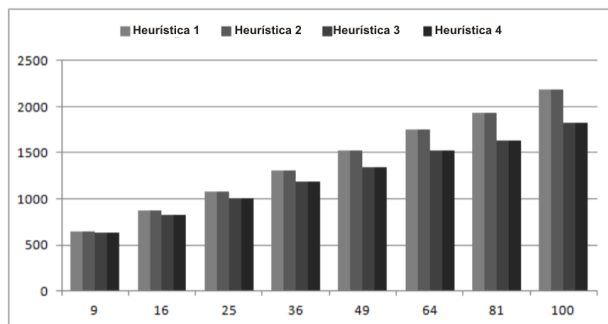
(b) Média de perda nos cenários com carga 1.



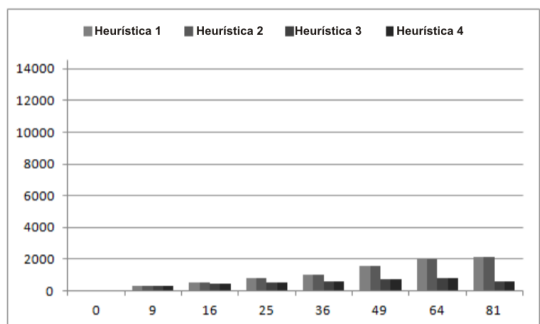
(c) Média de tempo gasto nos cenários com carga 2.



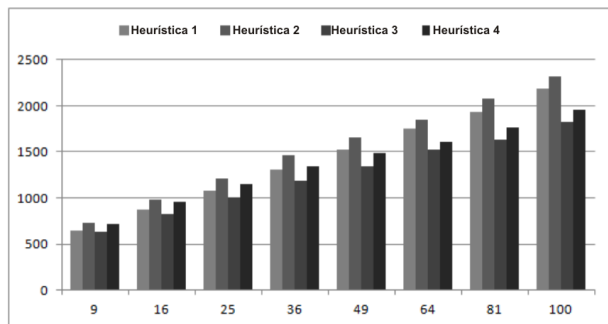
(d) Média de perda nos cenários com carga 2.



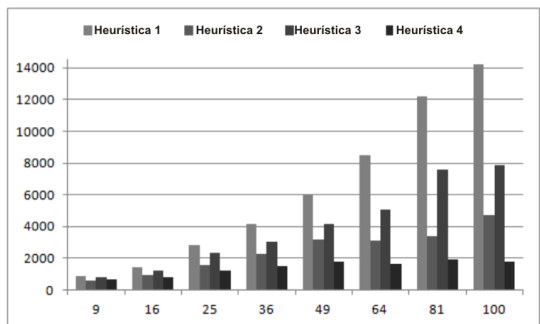
(e) Média de tempo gasto nos cenários com carga 3.



(f) Média de perda nos cenários com carga 3.



(g) Média de tempo gasto nos cenários com carga 4.



(h) Média de perda nos cenários com carga 4.

Figura 3. Gráficos apresentando as médias de tempo e perda de dados para cada um dos quatro tipos de carga.

Referências Bibliográficas

- [1] Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. e Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38:393–422.
- [2] Black, T.; Mak, V.; Pathirana, P. N. e Nahavandi, S. (2006). Using autonomous mobile agents for efficient data collection in sensor networks. *World Automation Congress (WAC), Budapest, Hungary*.
- [3] Di Francesco, M.; Das, S. K. e Anastasi, G. (2011). Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. *ACM Transactions on Sensor Networks*, 8(1).
- [4] Gandham, S. R.; Dawande, M.; Prakash, R. e Venkatesan, S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Global telecommunications conference, 2003. GLOBECOM'03. IEEE*, volume 1, p. 377381.
- [5] Kansal, A.; Somasundara, A. A.; Jea, D. D.; Srivastava, M. B. e Estrin, D. (2004). Intelligent fluid infrastructure for embedded networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services, MobiSys '04*, pp. 111–124, New York, NY, USA. ACM.
- [6] Lawler, E.; Lenstra, J. K.; Kan, A. H. G. R. e Shmoys, D. B. (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.
- [7] Loureiro, A. A. F.; Nogueira, J. M. S.; Ruiz, L. B. e Mini, R. A. (2002). Wireless sensors network (in portuguese). In *Proceedings of the 22nd Brazilian Computer Science Society Congress (SBC'02)*, pp. 193–234, Florianópolis, SC, Brazil. Tutorial.
- [8] Luo, J. e Hubaux, J.-P. (2005). Joint mobility and routing for lifetime elongation in wireless sensor networks. In *In Proceedings of IEEE INFOCOM*, pp. 1735–1746.
- [9] Nazário, G. A.; Iabrudi, A. e Ribeiro, S. (2012). Estudo do impacto da carga no planejamento de rota para coleta de dados por robôs móveis em redes de sensores sem fio. In *Proceedings of the Brazilian Conference on Intelligent Systems (BRACIS 2012)*.
- [10] Soares, M. B. (2011). *Um arcabouço estocástico para coleta de dados em RSSF utilizando um grupo de robôs móveis cooperativos*. PhD thesis, Universidade Federal de Minas Gerais.

- [11] Soares, M. B.; Campos, M. F. M.; Dutra, D. A.; Campos, V. C. d. S. e Pereira, G. A. S. (2007). Hybrid mobile robot navigational strategy for efficient data collection in sparsely deployed sensor networks. In *IROS'07*, pp. 2833–2838, San Diego, CA, USA.
- [12] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, pp. 254–265.
- [13] Somasundara, A.; Ramammorthy, A. e Srivastava, M. B. (2007). Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410.
- [14] Sugihara, R. e Gupta, R. K. (2010). Optimal speed control of mobile node for data collection in sensor networks. *IEEE Transactions on Mobile Computing*, 9(1):127–139.
- [15] Sugihara, R. e Gupta, R. K. (2011). Path planning of data mules in sensor networks. *ACM Transactions on Sensor Networks*, 8(1):1–27.
- [16] Yu, L.; Wang, N. e Meng, X. (2005). Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pp. 1214–1217.
- [17] Yuan, B.; Orlowska, M. e Sadiq, S. (2007). On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*.