

OTIMIZAÇÃO MULTIOBJETIVA POR ENXAME DE PARTÍCULAS DO PROTOCOLO DE ROTEAMENTO AODV EM REDES *AD HOC*

Trabalho de Conclusão de Curso

Engenharia de Computação

Aluno: Carlos Eduardo Buarque Cruz Pimentel

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho

Co-orientador: Prof. Dr. Renato Mariz de Moraes

Carlos Eduardo Buarque Cruz Pimentel

*Otimização Multiobjetiva por Enxame de
Partículas do Protocolo de Roteamento
AODV em Redes Ad Hoc*

Monografia apresentada para obtenção do Grau
de Bacharel em Engenharia de Computação
pela Universidade de Pernambuco

Orientador:

Prof. Dr. Carmelo José Albanez Bastos Filho

Co-orientador:

Prof. Dr. Renato Mariz de Moraes

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO
ESCOLA POLITÉCNICA DE PERNAMBUCO
UNIVERSIDADE DE PERNAMBUCO

Recife - PE, Brasil

dezembro de 2010

Resumo

As Redes de Telecomunicações, em especial as sem fio, estão desempenhando um papel cada vez mais importante e essencial nos dias atuais, seja no âmbito profissional ou no âmbito pessoal. Nesse cenário, as redes *ad hoc* destacam-se como uma excelente alternativa em situações onde nenhuma infraestrutura de comunicação existe. Entretanto, as características desse tipo de rede impõem uma série de desafios para seus projetistas.

Um dos pontos críticos em redes *ad hoc*, principalmente nas móveis, é o roteamento entre seus nós. Além de não existir um ponto central responsável por controlar toda a comunicação, a dinâmica topológica inerente a essas redes torna os enlaces das rotas ativas muito suscetíveis a falhas. Diversas são as abordagens que tentam realizar o roteamento em redes *ad hoc* considerando suas características particulares. Entre eles, destaca-se o *Ad Hoc On-Demand Distance Vector*, AODV, pelo seu amplo uso no estado-da-arte das pesquisas na área.

Diante da popularidade do AODV, muitos trabalhos basearam-se nele para sugerir propostas de otimização das redes *ad hoc*, inclusive através de técnicas de Computação Inteligente. Todavia, mesmo com uso de técnicas avançadas de computação, grande parte das propostas sugeridas só foca na otimização de um único parâmetro da rede, ou seja, procura realizar uma otimização de único objetivo.

No trabalho aqui apresentado, uma abordagem multiobjetiva foi desenvolvida para que a vazão e o atraso médio fossem otimizados simultaneamente em redes *ad hoc* que usam o AODV como o seu protocolo de roteamento. Para tanto, foi utilizada uma técnica de Computação Inteligente de otimização por enxame de partículas conhecida como *Multi Objective Particle Swarm Optimization - Crowding Distance and Roulette*, MOPSO-CDR.

Os resultados das simulações que implementaram a abordagem desenvolvida mostraram-se, para cenários com muita mobilidade, animadores quando comparados aos do AODV padrão e até mesmo aos de outras abordagens que usam o protocolo conjuntamente com técnicas de Computação Inteligente. Além dos resultados para as métricas que foram otimizadas pelo MOPSO-CDR, a vazão e o atraso médio da rede, foram obtidos resultados acerca de métricas secundárias que são a sobrecarga de pacotes de controle na rede e o número médio de saltos que um pacote percorre para chegar até o destino.

Para cenários com alta mobilidade e muitas conexões entre os nós, os resultados mostraram-se especialmente interessantes, com a vazão da rede chegando a 90% e o atraso médio reduzido até pela metade quando comparado com os resultados obtidos do AODV padrão e sendo também menor do que o atraso resultante do AODV usando outra técnica de Computação Inteligente. Para número médio de saltos, os valores obtidos equipararam-se com as demais abordagens, mas para a sobrecarga a diminuição chegou a até 40%.

Já para cenários com menos conexões entre os nós, como os resultados do AODV padrão já são bons para a vazão e o atraso médio, o emprego da nova abordagem não resultou em diferenças significativas para essas métricas. Contudo, para cenários com mobilidade total ou

quase total dos nós, a perda de pacotes caiu até pela metade. Ainda para esses cenários, a sobrecarga mostrou uma melhoria que chegou a até um $1/3$. O número de saltos, por sua vez, diminuiu, em média, 10%, considerando todos os cenários com poucas conexões simulados.

Com os resultados obtidos, o trabalho aqui desenvolvido é de excelente contribuição para promover um uso ainda maior de técnicas de Computação Inteligente no campo das Redes de Telecomunicações.

Abstract

The telecommunication networks, specially wireless ones, are playing a role increasingly important and essential nowadays, either in the professional or personal ambit. In this scenario, *ad hoc* networks stand out as an excellent alternative in situations where there is no communication infrastructure. However, the characteristics of this type of network impose a series of challenges to its designers.

One of the critical points in *ad hoc* networks, mainly mobile ones, is the routing among their nodes. Beyond the fact that there is no central point responsible for controlling the entire communication, the topological dynamics inherent to those networks makes the links of the routes more susceptible to breakages. Many approaches have tried to make the routing in *ad hoc* networks taking into consideration their own characteristics. Among these approaches, the Ad Hoc On-Demand Distance Vector routing protocol, AODV, stands out due to its broad use in the state-of-the-art research in the area.

Because of AODV protocol popularity, many works were based on it to present proposals for optimizing *ad hoc* networks, including the utilization of Intelligent Computational techniques for such goal. Nevertheless, even with the use of advanced techniques of computing, the majority of the proposals only focuses on optimizing one network parameter, i.e., they try to make a single objective optimization.

In the work presented here, a multiobjective approach was developed so that throughput and average delay were simultaneously optimized in *ad hoc* networks that use the AODV as their routing protocol. To do that, it was used an Intelligent Computational technique known as Multi Objective Particle Swarm Optimization - Crowding Distance and Roulette, MOPSO-CDR.

The results of the simulations that implemented the approach developed were, for scenarios with high mobility, encouraging when compared to the ones from default AODV protocol and even with those approaches using the protocol together with Intelligent Computational techniques optimizing only one network parameter. Beyond the results for the metrics that were optimized by MOPSO-CDR, the throughput and the average delivery delay of the network, results were obtained about secondary metrics like the overhead of control packets in the network and the average number of hops needed to a packet reach the destination.

For scenarios with high mobility and many connections among the nodes, the results were specially interesting, with network throughput of 90% and average delay reduced up to half when compared to the results obtained with default AODV protocol. Our approach also outperformed the AODV protocol using other Intelligent Computational technique. For the average number of hops, the values obtained are equivalent to the others approaches, but the overhead was reduced up to 40%.

In scenarios with fewer connections among nodes, as the results for the default AODV protocol are good for throughput and delay, the use of the new approach did not result in

significant differences for these metrics. However, for scenarios with high or moderate mobility of the nodes, the loss of packets halved. Still for such scenarios, the overhead was 1/3 lower. The number of hops, on the other hand, decreased, on average, 10%, considering every scenarios with fewer connections.

With the obtained results, the work developed here is an important contribution to promote an even further utilization of Intelligent Computational techniques in the telecommunication network area.

Agradecimentos

A conclusão do meu curso de graduação, representada neste trabalho, é a prova de que longas caminhadas só conseguem ser finalizadas com sucesso quando se tem ao seu redor pessoas maravilhosas.

Primeiramente, quero agradecer a Deus por, além de ter me dado a vida, ter sempre me capacitado para realizar todas as atividades da graduação e por colocar tantas pessoas maravilhosas e especiais no meu caminho.

Em especial, agradeço a minha mãe que, junto com meus avós, sempre foi a minha maior torcedora e apoiadora. Sem eles, com certeza, nada disso teria sido possível. Agradeço também aos familiares que sempre acreditaram no meu potencial.

Agradeço a minha namorada, Paula Morais, em quem encontrei também uma verdadeira companheira e que foi extremamente compreensiva e me deu muita força nesse período.

Como, sem amigos, ninguém é nada, não posso deixar de agradecer a todos os meus por continuarem ligando e se preocupando comigo mesmo que por muitas vezes eu não tenha atendido suas ligações e tenha ficado um bom tempo sem vê-los.

Dentro da Universidade, agradeço a todos os professores por tudo que me ensinaram e acrescentaram na minha vida. Em especial, agradeço ao Prof. Carmelo José Albanes Filho por ter sido um excelente orientador durante o desenvolvimento deste trabalho de conclusão de curso. Agradeço, principalmente, ao Prof. Renato Mariz de Moraes, meu orientador de Iniciação Científica e co-orientador deste trabalho, o qual foi fundamental com seus ensinamentos, postura e apoio para o êxito na conclusão da minha graduação.

Agradeço também aos amigos que fiz na Universidade, Leandro Honorato, Jefferson Amorim e Pércles Miranda, com os quais sempre pude contar e espero sempre poder contar.

Por fim, agradeço a Bruno Carlos Sales, Francisco Marinho Rodrigues e Marcos Antonio Oliveira Jr. por, mais do que colegas dos trabalhos da Universidade, serem meus amigos. Com eles aprendi e cresci muito.

Que venham as próximas caminhadas!

Sumário

Lista de Figuras	p. xi
Lista de Tabelas	p. xiii
Lista de Algoritmos	p. xiv
Lista de Abreviaturas e Siglas	p. xv
1 Introdução	p. 1
2 Redes <i>Ad Hoc</i>	p. 3
2.1 Roteamento nas Redes <i>Ad Hoc</i>	p. 4
2.1.1 <i>Ad Hoc On-Demand Distance Vector</i>	p. 6
3 Otimização Multiobjetiva por Enxame de Partículas	p. 11
3.1 Otimização Multiobjetiva	p. 12
3.1.1 Métricas de desempenho	p. 14
3.2 Otimização por Enxame de Partículas	p. 17
3.3 PSO Aplicado a Otimização Multiobjetiva	p. 20
4 Proposta e Arranjo Experimental	p. 23
4.1 AODV Utilizando o MOPSO-CDR	p. 24
4.2 Arranjo Experimental	p. 26
5 Resultados e Discussão	p. 29
5.1 Análise dos <i>Paretos Front</i>	p. 30

5.2	Análise dos resultados do AODV utilizando o MOPSO-CDR	p. 34
5.3	Comparação do AODV original com a versão PSO e a versão MOPSO-CDR	p. 36
6	Conclusões	p. 42
6.1	Trabalhos Futuros	p. 43
	Referências	p. 44

Lista de Figuras

1	Processo de envio do RREQ para descobrimento de uma rota.	p. 7
2	Relação de dominância para problema de dois objetivos.	p. 13
3	<i>Pareto front</i> para um exemplo de problema com dois objetivos.	p. 14
4	Ilustração com união de retângulos usados para o cálculo da métrica <i>hyper-volume</i> em um problema com dois objetivos.	p. 15
5	Vetores que representam as forças que influenciam o movimento de uma partícula.	p. 19
6	Exemplo do arquivo que guarda os valores do vetor de posição das partículas.	p. 24
7	Exemplo do arquivo que guarda o valor do <i>fitness</i> das partículas.	p. 25
8	Diagrama do funcionamento conjunto do AODV e MOPSO-CDR.	p. 25
9	Exemplo de arquivo externo gerado pelo MOPSO-CDR.	p. 27
10	<i>Pareto Front</i> dos cenários com área de 1500 x 300 metros e 10 conexões para as iterações 0, 100, 200 e 300.	p. 31
11	<i>Pareto Front</i> dos cenários com área de 1500 x 300 metros e 30 conexões para as iterações 0, 100, 200 e 300.	p. 32
12	Gráficos comparativos dos resultados das simulações para cenários com 10 conexões considerando as melhores soluções do MOPSO-CDR para atraso e vazão e a mais equilibrada.	p. 35
13	Gráficos comparativos dos resultados das simulações para cenários com 30 conexões considerando as melhores soluções do MOPSO-CDR para atraso e vazão e a mais equilibrada.	p. 36
14	Vazão em função do tempo de pausa do AODV original e do AODV utilizando o MOPSO-CDR para 10 conexões.	p. 37

15	Atraso médio em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.	p. 37
16	Número médio de saltos em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.	p. 38
17	Sobrecarga em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.	p. 39
18	Vazão em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.	p. 40
19	Atraso médio em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.	p. 40
20	Número médio de saltos em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.	p. 41
21	Sobrecarga em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.	p. 41

Lista de Tabelas

- 1 Tabela com os parâmetros escolhidos como melhores a partir da análise dos *Pareto Front* para cenários com 10 conexões. p. 33
- 2 Tabela com os parâmetros escolhidos como melhores a partir da análise dos *Pareto Front* para cenários com 30 conexões. p. 34

Lista de Algoritmos

1	Pseudocódigo da escolha do mecanismo de reparação de rota no AODV. . .	p. 9
2	Pseudocódigo da escolha do mecanismo de reparação de rota no AODV proposta em [Pereira 2010].	p. 10
3	Pseudocódigo do PSO.	p. 19
4	Pseudocódigo do MOPSO-CDR.	p. 22

Lista de Abreviaturas e Siglas

AODV - *Ad Hoc On-Demand Distance Vector*

CBR - *Constant bitrate*

CD - *Crowding Distance*

DSDV - *Destination Sequenced Distance Vector*

DSR - *Dynamic Source Routing*

ERB - *Estação Rádio Base*

IEEE - *Institute of Electrical and Electronics Engineers*

IoT - *Internet of Things*

MAC - *Media Access Control*

MANET - *Mobile Ad Hoc Network*

MOPSO - *Multiple Objective Particle Swarm Optimization*

MOPSO-CDLS - *Multiple Objective Particle Swarm Optimization - Crowding Distance and Local Search*

MOPSO-CDR - *Multiple Objective Particle Swarm Optimization - Crowding Distance and Roulette*

ns-2 - *The Network Simulator*

OLSR - *Optimized Link State Routing*

PAN - *Personal Area Network*

PSO - *Particle Swarm Optimization*

RERR - *Route Error*

RREP - *Route Reply*

RREQ - *Route Request*

RWP - *Random Waypoint*

TTL - *Time to live*

UDP - *User Datagram Protocol*

VANET - *Vehicle Ad Hoc Network*

Wi-Fi - *Wireless Fidelity*

WLAN - *Wireless Local Area Network*

WWAN - *Wireless Wide Area Network*

ZHLS - *Zone-based Hierarchical Link State*

ZRP - *Zone Routing Protocol*

1 *Introdução*

Redes *ad hoc* são redes nas quais as tarefas de administração e de controle da comunicação não são centralizadas em um único nó [Frodigh, Johansson e Larsson 2000]. Desta forma, todos os nós da rede, além de funcionarem como hospedeiros rodando aplicações, podem funcionar também como roteadores, ficando responsáveis por encaminhar até o destino as informações provenientes de seus vizinhos. O fato das redes *ad hoc* serem descentralizadas e não necessitarem de uma infraestrutura previamente montada, as torna excelentes para o uso em diversos cenários como monitoramento ambiental, comunicação em regiões de desastre, construção de PANs (do inglês, *Personal Area Networks*) e acesso de baixo custo à Internet. Entretanto, a falta de um controle central e a dinâmica da topologia impõem enormes desafios para os projetistas de tais redes, tornado-as um campo de pesquisa aberto e repleto de perguntas sem respostas.

Diante das características inerentes às redes *ad hoc*, uma área de muito interesse da comunidade científica é o roteamento nessas redes, já que o simples movimento de um nó participante de uma rota estabelecida e ativa pode implicar na quebra de um enlace de comunicação. Diversos protocolos de roteamento baseados em diferentes hipóteses já foram propostos [Murthy e Manoj 2004], [Broch et al. 1998], [Johnson e Maltz 1996], [Perkins e Royer 1997]. Entre esses protocolos, o AODV (do inglês, *Ad Hoc On-Demand Distance Vector*) é provavelmente o mais abordado na literatura [Perkins e Royer 1997].

O AODV pode realizar a reparação das eventuais falhas ocasionadas nas rotas estabelecidas a partir do nó fonte, chamada de reparação pela fonte, ou do nó predecessor à quebra de enlace, chamada de reparação local. Vários trabalhos apresentaram diferentes formas de otimizar a abordagem da recuperação de falhas de enlace do AODV, [Pan, Chuang e Wang 2005], [Youn et al. 2006], [Liu e Lin 2005], [Kai, Neng e Ai-fang 2005], [Noishiki, Yokota e Idoue 2005], [Pereira 2010]. Em [Pereira 2010], pesos foram atribuídos aos parâmetros usados na escolha do mecanismo de reparação e foi utilizado a Otimização por Enxame de Partículas (PSO, do inglês, *Particle Swarm Optimization*) [Kennedy e Eberhart 1995] para selecionar os valores desses pesos, de forma que o atraso médio da rede melhorasse.

Apesar do atraso médio ser um aspecto crítico de qualquer rede, ele não pode ser considerado isoladamente. Portanto, é interessante que a proposta considerada em [Pereira 2010] seja ampliada e aprofundada para considerar outros aspectos da rede e prover um estudo ainda mais valioso para o uso prático do AODV e, conseqüentemente, das redes *ad hoc* móveis.

Este trabalho de conclusão de curso visou considerar a otimização do atraso médio em conjunto com a da vazão, outro aspecto fundamental de redes de comunicação em geral, inclusive das redes *ad hoc*. Dessa forma, uma otimização multiobjetiva por enxame de partículas, através do algoritmo MOPSO-CDR (do inglês, *Multi Objective Particle Swarm Optimization - Crowding Distance and Roulette*) [Santana 2009], será realizada para que seja possível minimizar simultaneamente o atraso da rede e a vazão.

Cenários de simulação foram construídos adotando valores utilizados na literatura referente ao estudo de redes *ad hoc* para parâmetros como área, número total de nós e número total de conexões. Esses cenários foram simulados utilizando o *Network Simulator*, o ns-2 [The Network Simulator - ns-2], e os resultados foram comparados aos obtidos em [Pereira 2010] em termos de atraso e vazão. Apesar da análise primordial e da otimização de fato serem realizadas com esses aspectos, outros aspectos, como sobrecarga e número médio de saltos, também foram analisados.

Utilizar uma abordagem multiobjetiva permite obter resultados mais valiosos para o entendimento do impacto da aplicação de Inteligência Computacional, especificamente da técnica PSO, em protocolos de roteamento de redes *ad hoc*, no caso o AODV, uma vez que, na prática, busca-se o desenvolvimento de protocolos de redes que sejam eficientes em diversos aspectos e não apenas em um.

Os demais capítulos deste documento organizam-se da seguinte forma: o capítulo 2 detalha os conceitos referentes às redes *ad hoc*, ao roteamento nessas redes e ao protocolo AODV. O capítulo 3 inicia-se com uma explanação da ideia de otimização multiobjetiva, continua com a apresentação geral dos fundamentos da PSO e termina mostrando como essa técnica de Inteligência Computacional é utilizada para solucionar problemas com múltiplos objetivos conflitantes. O capítulo 4 apresenta o detalhamento da proposta do trabalho de conclusão de curso, de como foi realizada a integração entre o MOPSO-CDR e o AODV e do ambiente de simulação montado para obter os resultados. Por sua vez, os resultados e as discussões acerca deles são mostrados no capítulo 5. As conclusões e os trabalhos futuros, por fim, são apresentados no capítulo 6.

2 *Redes Ad Hoc*

Redes de comunicação sem fio têm se tornado cada vez mais ubíquas e têm sido amplamente utilizadas por um número crescente de pessoas para diversas finalidades. Dentre os tipos de redes sem fio mais utilizadas atualmente, estão as redes celulares e as redes Wi-Fi. Apesar das redes Wi-Fi serem destinadas para acesso sem fio local, ou seja, serem do tipo WLAN (do inglês, *Wireless Local Area Network*), e redes celulares procurarem prover acesso sem fio e móvel a grandes áreas, sendo redes do tipo WWAN (do inglês, *Wireless Wide Area Network*), ambas compartilham da necessidade de uma infraestrutura previamente instalada para que possam funcionar. Nas redes Wi-Fi, essa infraestrutura fica a cargo dos pontos de acesso e nas celulares, fica por conta das ERBs, as estações rádio-base.

Com a ampla utilização dessas redes e com a rápida evolução das tecnologias por elas utilizadas, a tendência é que elas se tornem ainda mais difundidas e mais utilizadas nos próximos anos. Apesar do sucesso dessas redes, existe uma série de aplicações e de cenários cujo o uso de redes com infraestrutura é inapropriado. É para essas aplicações e cenários que foram desenvolvidas as redes *ad hoc* [Frodigh, Johansson e Larsson 2000].

As redes *ad hoc* têm suas origens datadas de 1968, quando foi iniciado o projeto da rede ALOHA, cujo objetivo era interconectar as escolas do estado do Havaí, nos Estados Unidos [Abramsson 1970]. A rede ALOHA foi uma rede de dispositivos fixos que utilizava um protocolo distribuído de controle de acesso ao canal de comunicação, tornando-se a base para o desenvolvimento posterior das redes *ad hoc*.

Nessas redes, toda a gerência da comunicação é de responsabilidade dos nós que as compõem. Dessa forma, os nós de uma rede *ad hoc*, além de funcionarem como fonte ou destino das conexões na rede, realizam tarefas de administração e de controle da comunicação de forma descentralizada. Tal característica faz com que redes desse tipo sejam excelentes para ambientes dinâmicos, onde há a necessidade de comunicação entre os dispositivos de rede sem a configuração prévia de uma estrutura de apoio.

As aplicações para as redes *ad hoc* vão desde comunicação de curta distância e pouca

ou nenhuma mobilidade, como a transferência de arquivos entre celulares e *notebooks*, até comunicação em grandes áreas e com alta mobilidade, como em situações de desastre. Os exemplos abaixo mostram a versatilidade das redes *ad hoc*.

- **No trabalho:** o usuário chega a sua sala e automaticamente tem seus arquivos e outros dados relevantes sincronizados entre seu *smartphone* e computador, além de poder enviar diretamente arquivos do seu *smartphone* para o *smartphone* de um colega.
- **Em casa:** o usuário pode ter a sua disposição robôs domésticos capazes de se comunicar para saber quando entrar em ação e, caso tenha problemas de saúde, pode contar com dispositivos de monitoramento que se comunicam entre si e com centros médicos para relatar qualquer anomalia. Este cenário ilustra bem um novo paradigma das redes de comunicação: a Internet das Coisas ou IoT (do inglês, *Internet of Things*) [Atzori, Iera e Morabito 2010];
- **Em veículos:** no cenário da comunicação interveicular, as redes *ad hoc* são chamadas de VANETs (do inglês, *Vehicle Ad Hoc Networks*) [Yousefi, Mlousavi e Fathy 2006], uma subclasse das redes *ad hoc* móveis, as MANETs (do inglês, *Mobile Ad Hoc Networks*). Nesse cenário, além de contar com acesso à Internet sem depender das redes celulares, os usuários podem contar com um novo sistema de navegação onde os carros transferem as informações entre si sobre o tráfego e postos de gasolina nas rotas, por exemplo.

Apesar da versatilidade das redes *ad hoc*, o controle e a administração da comunicação descentralizados impõem uma série de desafios aos seus projetistas. No caso das redes *ad hoc* móveis, as MANETs, esses desafios são ainda maiores, já que, a mobilidade dos nós é um fator de dificuldade a mais no roteamento dos pacotes da rede, por exemplo.

2.1 Roteamento nas Redes Ad Hoc

No início do desenvolvimento das redes *ad hoc*, no caso da rede ALOHA, um nó fonte só poderia enviar pacotes de dados para um destino se este estivesse no seu raio de cobertura, sendo, portanto, exclusivamente uma rede de único salto. Nesse modelo, a conectividade fica limitada, uma vez que um nó não é capaz de se comunicar com outros nós da mesma rede caso eles estejam fora do seu raio de alcance.

Com o posterior desenvolvimento da PRnet pela Defense Advanced Research Projects Agency, a DARPA dos EUA, [J.Jubin e J.D.Tornow 1987], o conceito de múltiplos saltos foi

introduzido no contexto das redes *ad hoc*. Em uma rede *ad hoc* de múltiplos saltos, os nós da rede funcionam como roteadores, cooperando uns com os outros na transmissão dos pacotes de dados, fazendo com que um nó consiga se comunicar com nós fora do seu alcance de transmissão, aumentando, assim, a conectividade da rede. Em um cenário como esse, o roteamento da rede passa a ser uma atividade imprescindível e de fundamental importância.

Entretanto, por causa da dinâmica da topologia inerente às redes *ad hoc*, estabelecer um caminho pelo qual os pacotes de dados serão roteados não é uma tarefa fácil. A dificuldade torna-se ainda maior em MANETs, uma vez que o simples movimento, para outro local da rede, de um nó participante de uma rota ativa de comunicação pode implicar na quebra de um enlace de comunicação. Uma vez que um enlace foi quebrado, é necessário que uma nova rota seja estabelecida o mais rápido possível, evitando o desperdício de recursos muita vezes escassos na rede como largura de banda de transmissão e energia.

Os protocolos de roteamento desenvolvidos para as redes *ad hoc* podem ser classificados de acordo com a forma utilizada na descoberta das rotas, sendo: pró-ativos, sob-demanda e híbridos. Os detalhes dessa classificação são descritos a seguir.

- **Pró-ativos:** São os protocolos orientados à tabela de roteamento, pois matêm rotas para todos os nós, até mesmo para os nós que não estão recebendo nenhum pacote. De maneira similar aos protocolos de roteamento das redes cabeadas, esses protocolos constantemente enviam pacotes de controle para a rede com a finalidade de monitorar qualquer mudança na sua topologia e, assim, agir de maneira rápida no estabelecimento de rotas. Quanto maior for a dinamicidade da rede, maior deve ser a troca dessas mensagens de controle, o que afeta diretamente o consumo de banda e de energia. São exemplos de protocolos de roteamento pró-ativos o *Destination Sequenced Distance Vector (DSDV)* [Perkins e Bhagwat 1994] e o *Optimized Link State Routing (OLSR)* [Jacquet et al. 2001].
- **Sob-demanda:** Também conhecidos como reativos, esses protocolos são orientados à demanda de roteamento, pois as rotas entre os nós são determinadas apenas quando há, de fato, a necessidade de troca de pacotes entre eles. Dessa forma, não há uma troca periódica de mensagens, uma vez que a troca dessas mensagens é disparada apenas no evento de envio de dados por parte de um nó. Essa abordagem permite que os nós foquem apenas nas rotas sendo utilizadas e naquelas ainda sendo estabelecidas e não em todas as rotas possíveis da rede. Apesar da economia de recursos da rede obtidas com o uso desses protocolos, há um aumento no tempo em que um nó leva para começar a enviar seus pacotes à rede. O *Dynamic Source Routing (DSR)* [Johnson e Maltz 1996]

e o *Ad Hoc On-Demand Distance Vector* (AODV) [Perkins e Royer 1997] são bons exemplos de protocolos sob-demanda.

- **Híbridos:** Esses protocolos tentam reunir o que há de melhor nos protocolos pró-ativos e nos reativos, tornando-se boa opção para redes cujo comportamento varia ao longo do tempo. Nos protocolos híbridos, cada nó define uma área de serviço, chamada de zona, e atua de forma pró-ativa dentro dessa zona na descoberta de rotas para nós que também encontram-se nessa área. A atuação reativa ocorre no caso onde deseja-se descobrir rotas entre nós de diferentes zonas. Os protocolos híbridos são escaláveis em relação aos pró-ativos à medida que o número de nós aumenta, pois não enviam mensagens de controle constantemente para toda a rede. Entretanto, em redes grandes, a latência no estabelecimento de uma rota é um problema, assim como acontece com os protocolos sob-demanda. Exemplos desses protocolos são o *Zone Routing Protocol* (ZRP) [Haas e Pearlman 2000] e o *Zone-based Hierarchical Link State* (ZHLS) [Joa-Ng e Lu 1999].

O SwitchLab, da Ericsson Research, realizou um estudo para comparar um protocolo pró-ativo, o DSDV, com dois sob-demanda, o AODV e o DSR, [Johansson et al. 1999], e observou que os protocolos sob-demanda têm melhor desempenho do que o protocolo pró-ativo, em termos de atraso e vazão. Além disso, os protocolos sob-demanda apresentaram melhor utilização dos recursos da rede. O AODV e o DSDV, quando comparados entre si, apresentaram desempenho similar. Diante desses resultados e do amplo uso do AODV na literatura, esse foi o protocolo escolhido para o desenvolvimento do trabalho aqui apresentado.

2.1.1 *Ad Hoc On-Demand Distance Vector*

O *Ad Hoc On-Demand Distance Vector*, AODV [Perkins e Royer 1997], é um protocolo de roteamento para o uso em redes *ad hoc* móveis que permite um roteamento de múltiplos saltos, dinâmico e auto-inicializável entre os nós da rede que desejem estabelecer e manter uma conexão entre si [Perkins, Belding-Royer e Das 2003]. Por ser um protocolo sob-demanda, ou seja, reativo, o AODV torna possível o estabelecimento relativamente rápido de novas rotas sem a necessidade de conhecer previamente as rotas entre todos os nós, inclusive entre aqueles que não estão ativos.

O AODV foi desenvolvido para operar estabelecendo rotas livres de laços, ou *loops*, e com isso responder de maneira eficiente às quebras de enlaces e às mudanças de topologia, tanto em redes com baixa mobilidade quanto em redes com mobilidade alta, nos mais variados níveis

de tráfego. Essa operação sem laços é garantida pelo uso do número de sequência do destino das rotas armazenadas na tabela de roteamento dos nós da rede. Esse número de sequência é criado pelo nó destino e enviado conjuntamente com qualquer informação sobre a rota para os nós que a requisitem. Se um nó tiver duas opções de rotas para um destino, ele vai escolher aquela cujo número de sequência é maior.

Apenas quando um nó precisa descobrir uma rota até um novo destino é que o AODV entra em ação, através da difusão de mensagens de requisição por parte desse nó. Essas mensagens usadas na descoberta de uma rota até um novo destino são chamadas de RREQs (do inglês, *Route Requests*). Quando um RREQ atinge o nó destino ou um nó intermediário que possui uma rota suficientemente nova até o destino, a rota pode ser então determinada. Uma rota suficientemente nova significa uma entrada na tabela de roteamento de um nó, até o destino desejado, que tem atrelado a si um número de sequência maior ou igual àquele contido no RREQ. Se, ao receber um RREQ, o nó não for o destino desejado nem possuir uma rota nova o suficiente até o destino, ele trata de repassar o RREQ para seus vizinhos até que a rota seja descoberta. A Figura 1 mostra esse processo e identifica o número de sequência da entrada na tabela de roteamento como NS_E e o número de sequência contido no RREQ como NS_RREQ.

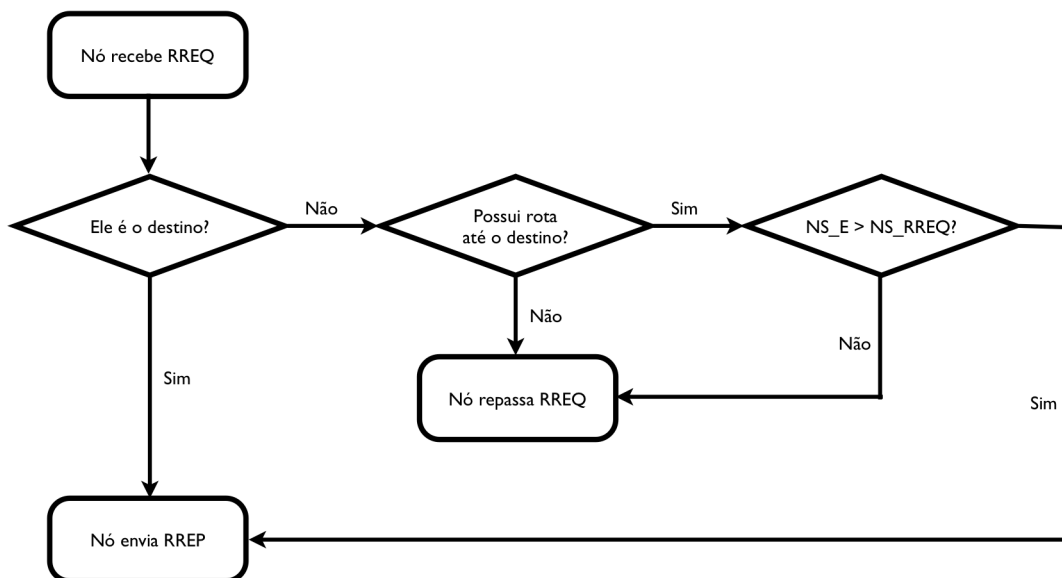


Figura 1: Processo de envio do RREQ para descobrimento de uma rota.

Quando o RREQ atinge um nó que é o destino ou que tem na sua tabela de roteamento uma entrada que corresponda ao destino desejado e não seja antiga, esse nó dispara uma mensagem chamada de RREP (do inglês, *Route Reply*) diretamente à origem da requisição de rota. Esse envio de volta é possível uma vez que cada nó, ao receber o RREQ, guarda uma rota de volta até o nó que originou a mensagem. A rota torna-se ativa ou disponível quando

o RREP chega ao nó fonte da conexão a ser estabelecida.

Tornando-se ativa, uma rota passa a ser monitorada pelos nós que fazem parte dela para identificar e recuperar eventuais quebras de enlaces resultantes da dinâmica da topologia da rede. O AODV usa um dos seguintes métodos para detectar quebras de enlaces: informações a partir da própria camada de enlace ou uso de mensagens de controle conhecidas como mensagens *hello*. Diante da dificuldade em se obter informações a partir da camada de enlace, o uso das mensagens *hello* é mais utilizado na análise da conectividade das redes *ad hoc* que utilizam o AODV como protocolo de roteamento [Chakeres e Belding-Royer 2002].

As mensagens *hello* nada mais são do que RREPs com o valor do seu *time to live*, TTL¹, igual a 1, ou seja, só os vizinhos de um nó que envia uma mensagem *hello* a receberão. Essas mensagens só são enviadas por um nó se ele for parte de uma rota ativa e se, em um determinado período tempo, ele não tiver enviado nenhuma mensagem por difusão (por exemplo, um RREQ). O não recebimento por um nó de nenhuma mensagem de difusão ou de nenhuma mensagem *hello* de um vizinho, por um certo período de tempo, indica que o enlace entre ele e aquele vizinho foi quebrado.

Quando ocorre a quebra de um enlace em uma rota ativa, o AODV tentará realizar a recuperação da falha através de uma das duas abordagens: reparação local ou reparação pela fonte. Optando-se por realizar a reparação local, o nó predecessor à quebra de enlace tentará reestabelecer o trecho da rota que parte dele e vai até o destino, tornando todo o processo transparente para o nó fonte. Caso o nó predecessor não consiga o reestabelecimento do trecho, ele enviará para a rede, através de difusão, uma mensagem chamada de RERR (do inglês, *Route Error*). Esta mensagem poderá chegar diretamente ao nó fonte, caso esse esteja na área de alcance do nó predecessor, ou através do seu reenvio por parte dos outros nós da rede. Uma vez que o nó fonte recebe o RERR, ela iniciará um novo processo de descobrimento de rota até o nó para o qual ele quer enviar pacotes de dados, realizando, assim, uma reparação pela fonte.

A escolha de qual mecanismo de recuperação será empregado depende do número de saltos já percorridos pelos pacotes até chegar ao nó predecessor e do número de saltos que separa esse nó do nó destino. Caso o número de saltos já percorridos por um pacote seja maior do que o número de saltos restantes, a reparação local é escolhida; caso contrário, a reparação pela fonte é realizada. O Algoritmo 1 mostra o pseudocódigo dessa tomada de decisão.

Estudos realizados em [Pereira, Buarque e Moraes 2009] mostraram que essa abordagem

¹Em termos de Redes de Computadores, o *time to live*, ou TTL, é um termo do inglês usado para designar por quantos saltos ou período de tempo um pacote será propagado pela rede

Algoritmo 1: Pseudocódigo da escolha do mecanismo de reparação de rota no AODV.

Entrada: *numeroSaltosPercorridos* e *numeroSaltosRestantes*

1 **início**

2 **se** *numeroSaltosPercorridos* > *numeroSaltosRestantes* **então**

3 | *reparoLocal()*;

4 **senão**

5 | *reparoFonte()*;

6 **fim**

híbrida, que considera ora o mecanismo de reparação pelo nó predecessor à quebra ora o de reparação pela fonte, nem sempre é a mais eficiente. Dependendo da topologia e dos parâmetros da rede e das aplicações dos usuários, escolher por padrão um mecanismo de recuperação em relação ao outro pode ser preferível.

Em [Pereira 2010], buscou-se manter a flexibilidade de escolha apresentada no Algoritmo 1, padrão do AODV, mas a conectividade tanto do nó predecessor à quebra quanto a do nó fonte passaram a ser consideradas na hora de eleger um mecanismo de reparação para ser utilizado, com a intenção de que melhores escolhas sejam feitas. As modificações realizadas por [Pereira 2010] no algoritmo do AODV são mostradas no Algoritmo 2.

Buscando melhorar ainda mais o desempenho do protocolo AODV, ainda em [Pereira 2010], pesos foram adicionados a cada um dos parâmetros usados para decidir qual mecanismo de reparação de rota deve ser usado diante de uma quebra de enlace. Entretanto, para que os valores escolhidos para esses pesos fossem os melhores possíveis em relação à diminuição do atraso médio na rede, a técnica de Inteligência Computacional conhecida por Otimização por Enxame de Partículas foi utilizada.

Os resultados obtidos a partir da aplicação dessa técnica de otimização ao AODV foram interessantes não só em relação ao atraso médio da rede, métrica que foi objetivo da otimização, mas também em relação a vazão, número de pacotes de controle na rede (sobrecarga) e número médio de saltos que um pacote percorre até chegar ao destino. Tais resultados motivaram a extensão desse trabalho, através da otimização, aqui proposta, de mais aspectos de uma rede *ad hoc*. Por motivos que serão abordados no próximo capítulo, preferiu-se utilizar uma variação recente da otimização multiobjetiva por enxames de partículas.

Algoritmo 2: Pseudocódigo da escolha do mecanismo de reparação de rota no AODV proposta em [Pereira 2010].

Entrada: *numeroSaltosPercorridos*, *numeroSaltosRestantes*,
conectividadeFonte, *conectividadePredecessor* e pesos *A*, *B*, *C* e *D*

```
1 início
2   fonte = A * numeroSaltosPercorridos + B * conectividadeFonte;
3   local = C * numeroSaltosRestantes + D * conectividadePredecessor;
4   se fonte > local então
5     | reparoLocal();
6   senão
7     | reparoFonte;
8 fim
```

3 Otimização Multiobjetiva por Enxame de Partículas

Diversos problemas, nas mais variadas áreas do conhecimento, têm que lidar com a otimização de mais de um objetivo de forma conjunta, sendo estes chamados de problemas de otimização multiobjetiva. Em muitos casos, os objetivos a serem otimizados são conflitantes e a otimização multiobjetiva do problema produzirá, diferentemente da otimização de único objetivo, não apenas uma solução, mas um conjunto de soluções ótimas que buscam satisfazer as diferentes necessidades consideradas.

Dentre as diversas áreas que necessitam da otimização multiobjetiva na resolução de seus problemas, pode-se destacar a área de Redes de Telecomunicações. Com o uso cada vez maior dessas redes para a transferência de diversos tipos de informações e dados, tanto no âmbito pessoal quanto no profissional, otimizá-las considerando vários aspectos é de suma importância. Nos casos em que há um maior controle sobre a infraestrutura da rede, essa otimização, apesar de não ser trivial, torna-se mais simples do que nos casos onde o controle é mínimo ou nulo, como nas redes celulares e nas redes *ad hoc*. Essa dificuldade surge da mobilidade inerente dos nós, tanto nas redes celulares quanto nas redes *ad hoc* do tipo móvel, as MANETs, e torna-se ainda maior nas redes *ad hoc*, de maneira geral, devido ao caráter de “improvisado” delas [Frodigh, Johansson e Larsson 2000].

Pelas dificuldades que as características das redes *ad hoc* impõem e diante da gama de cenários em que elas podem ser usadas, otimizar o desempenho dessas redes é um problema abordado em diversos trabalhos como [Gupta et al. 2010] e [Ma e Kim 2009]. Boa parte desses trabalhos busca a otimização através de novas propostas de protocolos de roteamento [Perkins e Bhagwat 1994], [Perkins e Royer 1997], [Haas e Pearlman 2000], ou da melhoria de protocolos desse tipo já existentes [Pan, Chuang e Wang 2005], [Liu e Lin 2005], [Pereira 2010]. Apesar das propostas apresentadas serem desenvolvidas tendo em vista a otimização de certos aspectos da rede, como uso de energia, atraso e sobrecarga, essa nem sempre é a meta principal e quando é, geralmente apenas um aspecto é considerado [Pereira 2010].

Considerando a complexidade dos problemas de otimização multiobjetiva e a importância das áreas a que eles pertencem, como é o caso da melhoria da performance de redes *ad hoc* em termos de roteamento, diversas abordagens de otimização multiobjetiva foram desenvolvidas. Entre essas abordagens, destacam-se as que fazem utilização da Inteligência Computacional, como a otimização por enxame de partículas, por exemplo.

A Otimização por Enxame de Partículas, conhecida como PSO (do inglês, *Particle Swarm Optimization*) [Kennedy e Eberhart 1995] é uma técnica bastante popular usada tanto em problemas de busca quanto em problemas de otimização, mas que, na sua versão original, só é capaz de realizar otimização de único objetivo. Entretanto, graças a sua simplicidade, seu bom desempenho e sua alta capacidade de convergência, diversos trabalhos procuraram estendê-la para que ela também fosse aplicada a problemas de otimização multiobjetiva [Coello, Pulido e Lechuga 2004], [Tsou et al. 2007], [Santana 2009]. Dessa forma, a PSO, na sua forma multiobjetiva, tornou-se a candidata natural e mais adequada para ser utilizada conjuntamente com o AODV para a otimização de redes *ad hoc*.

3.1 Otimização Multiobjetiva

A otimização multiobjetiva busca encontrar soluções capazes de satisfazer os objetivos, muitas vezes conflitantes, de um problema em questão. Dessa forma, pode-se escrever, de maneira genérica, um problema de otimização multiobjetiva, com restrições a serem satisfeitas, da seguinte forma [Coello, Pulido e Lechuga 2004]:

otimizar

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})], \quad (3.1)$$

sujeito a restrições dadas pela inequação

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \dots, m, \quad (3.2)$$

e pela equação

$$h_j(\vec{x}) = 0 \quad j = 1, 2, \dots, p, \quad (3.3)$$

onde $\vec{x} = [x_1, x_2, \dots, x_n]$ é o vetor de variáveis de decisão, $f_i : R^n \rightarrow R, i = 1, \dots, k$ são as funções objetivo e $g_i, h_j : R^n \rightarrow R, i = 1, \dots, m, j = 1, \dots, p$ são as funções de restrição do problema.

Algumas das soluções encontradas formam um conjunto no qual elas são equivalentes

entre si e consideradas superiores às demais, ou seja, os resultados apresentados por elas para cada objetivo são iguais ou melhores que os resultados das demais soluções. Estas soluções são chamadas de não dominadas ou soluções do *Pareto-ótimo*.

Para facilitar o entendimento do conceito de dominância entre soluções de um problema de otimização multiobjetiva, algumas definições são construídas com base em um problema de minimização [Reyes-Sierra e Coello 2006].

Definição 1 Dados dois vetores $\vec{x}, \vec{y} \in R^n$, considera-se que $\vec{x} \leq \vec{y}$ se $x_i \leq y_i$ para $i = 1, \dots, k$ e que \vec{x} domina \vec{y} (definido como $\vec{x} \prec \vec{y}$) se $\vec{x} \leq \vec{y}$ e $\vec{x} \neq \vec{y}$.

A Figura 2 mostra um caso de relação de dominância para um problema de dois objetivos.

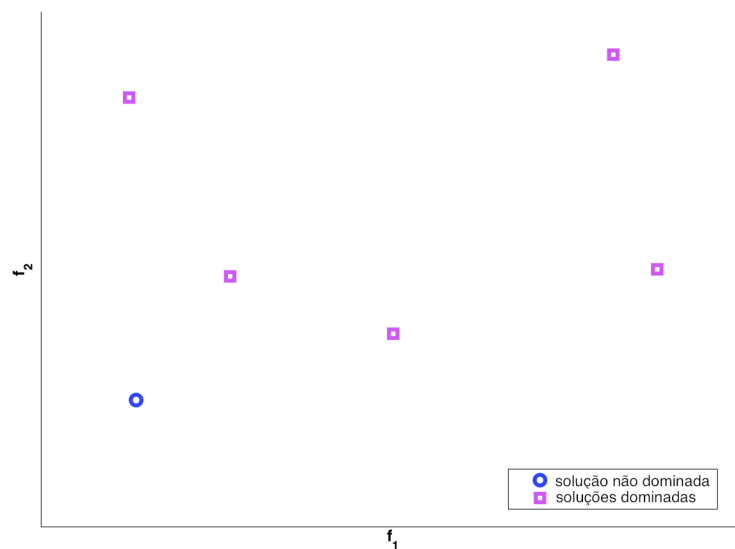


Figura 2: Relação de dominância para problema de dois objetivos.

Definição 2 Um vetor de variáveis de decisão $\vec{x} \in \chi \subset R^n$ é não dominado em relação a χ se não existir outro $\vec{x}' \in \chi$ de tal forma que $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$.

Definição 3 Um vetor de variáveis de decisão $\vec{x}^* \in F \subset R^n$ (F sendo uma região alcançável) é considerado Pareto-ótimo se não é dominado em relação a F .

Definição 4 O conjunto de soluções do Pareto-ótimo P^* é definido por:

$$P^* = \{ \vec{x} \in F | \vec{x} \text{ é Pareto-ótimo} \}.$$

Definição 5 O *Pareto Front* PF^* é definido por:

$$PF^* = \{\vec{f}(\vec{x}) \in R^k | \vec{x} \in P^*\}.$$

A Figura 3 mostra o *Pareto front* para um problema de dois objetivos.

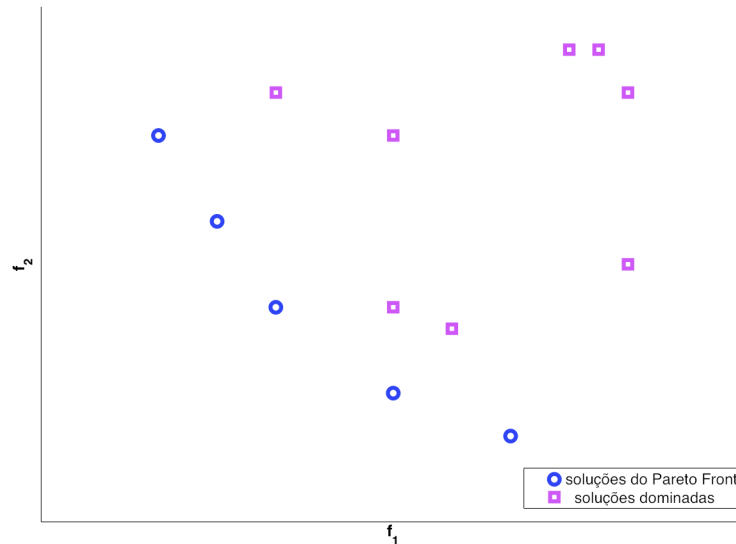


Figura 3: *Pareto front* para um exemplo de problema com dois objetivos.

Como, matematicamente, o *Pareto Front* é a imagem do conjunto de soluções do *Pareto-ótimo* no espaço de objetivos [Nedjah, Coelho e Mourelle 2010], a intenção da otimização multiobjetiva é determinar o conjunto de soluções do *Pareto-ótimo* a partir do conjunto F de todos os vetores de variáveis de decisão que satisfazem as equações (3.2) e (3.3), de modo que este se assemelhe com o *Pareto Front* esperado. Vale notar que, na prática, nem todo conjunto de soluções do *Pareto-ótimo* é normalmente o desejável ou alcançável.

Durante a busca das soluções para um problema, a otimização multiobjetiva tenta manter a diversidade, ou seja, tenta encontrar um conjunto de soluções que sejam bem distribuídas entre os objetivos.

3.1.1 Métricas de desempenho

A otimização multiobjetiva possui algumas métricas que são capazes de quantificar e qualificar o *Pareto Front* obtido. Cada métrica analisa um aspecto diferente do *Pareto Front* e entre as mais usadas estão *hypervolume*, *spacing*, *coverage* e *maximum spread*.

Hypervolume

Proposta por [Zitzler 1999], essa métrica é definida pela área do espaço de busca do objetivo ocupada pelo *Pareto Front*, ou seja, é a área embaixo da curva.

Em um problema de otimização com dois objetivos, considerando que cada ponto $(f_1(\vec{x}), f_2(\vec{x}))$ pertencente ao *Pareto Front*, juntamente com a origem, delimita um retângulo no espaço dos objetivos, a métrica *hypervolume* corresponde à área formada pela união de todos os retângulos, como mostrado na Figura 4.

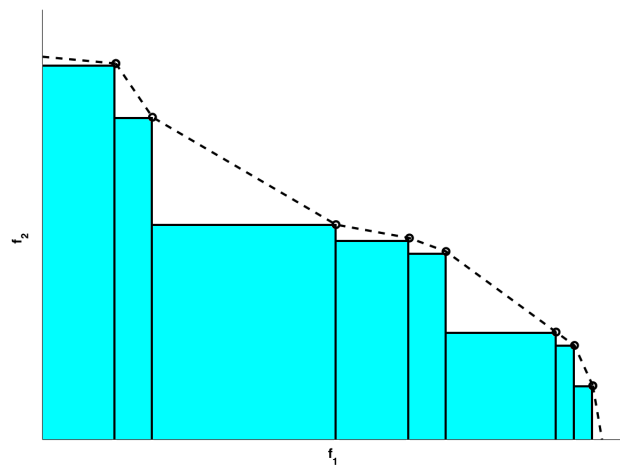


Figura 4: Ilustração com união de retângulos usados para o cálculo da métrica *hypervolume* em um problema com dois objetivos.

A equação (3.4) generaliza a ideia para problemas de n -objetivos:

$$HV = \left\{ \bigcup_i a_i | x_i \in P_a^* \right\}, \quad (3.4)$$

onde x_i é um vetor de soluções não dominadas contido em P_a^* e a_i é a métrica *hypervolume* determinada pelos componentes de x_i e a origem.

Spacing

A métrica *spacing* pode ser usada para analisar como as soluções não dominadas distribuem-se pelo *Pareto Front*. Em outras palavras, essa métrica estima a variância da distância entre soluções não dominantes adjacentes, como pode ser visto pela Equação (3.5):

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}, \quad (3.5)$$

onde $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$, $i, j = 1, \dots, n$, \bar{d} é a distância entre todas as soluções adjacentes e n é o número de soluções não dominadas do *Pareto Front*.

Coverage

Métrica proposta por [Zitzler 1999] e que mapeia o par ordenado (A, B) para o intervalo $[0, 1]$ usando a seguinte equação:

$$C(A, B) = \frac{|\{b \in B; \exists a \in A : a \succeq b\}|}{|B|}. \quad (3.6)$$

$C(A, B)$ com um valor de 1 significa que todas as soluções em B são fracamente dominadas por A . Por outro lado, se o valor de $C(A, B)$ for 0, nenhuma solução em B é fracamente dominada por A .

Vale salientar que é necessário calcular $C(A, B)$ e $C(B, A)$, uma vez que $C(A, B)$ não é necessariamente igual a $1 - C(B, A)$.

No caso de $0 < C(A, B) < 1$ e $0 < C(B, A) < 1$, nem A domina fracamente B , nem B domina fracamente A . Tal situação significa que os conjuntos A e B são incomparáveis, ou seja, A não é pior que B e vice-versa.

Maximum spread

Proposta também por [Zitzler 1999], essa métrica calcula a extensão máxima que as soluções não dominadas no *Pareto* abrangem.

Considerando um problema com dois objetivos, a métrica *maximum spread* vai ter seu valor obtido da aplicação direta da distância Euclidiana entre as duas soluções mais distantes.

$$MS = \sqrt{\sum_{m=1}^M (\max_{i=1}^n f_m^i - \min_{i=1}^n f_m^i)^2}, \quad (3.7)$$

onde n é o número de soluções no *Pareto Front* e M é o número de objetivos do problema. Vale salientar que quanto maiores os valores, melhor o desempenho.

3.2 Otimização por Enxame de Partículas

A Otimização por Enxame de Partículas, também conhecida como PSO (do inglês, *Particle Swarm Optimization*) [Kennedy e Eberhart 1995], é uma técnica de Inteligência Computacional baseada no comportamento de bandos de pássaros na busca por alimento. Descoberto a partir da simulação de modelos sociais simplificados, a PSO é geralmente utilizada para otimização em problemas cujas funções são não lineares e as variáveis são contínuas. Sua principal inspiração é a teoria de enxames. A PSO tornou-se uma técnica bastante popular e, segundo [Reyes-Sierra e Coello 2006], essa popularidade deve-se principalmente a dois fatores-chave:

1. O algoritmo do PSO é simples, o que torna sua implementação também simples. Além disso, há uma variedade muito grande de códigos fonte do algoritmo PSO disponíveis em domínio público [Particle Swarm Optimization: Source Codes & Demos].
2. Para uma grande variedade de problemas, a PSO tem se mostrado muito eficiente, produzindo excelentes resultados a um custo computacional muito baixo.

No enxame definido pelo algoritmo do PSO, as partículas que o compõem comunicam-se entre si para trocar o conhecimento que cada uma adquiriu e, dessa forma, tornam-se capazes de tomar decisões baseadas não apenas no seu conhecimento, mas também no conhecimento do grupo (com as partículas representando os pássaros e o enxame, por sua vez, o bando).

Em sua forma mais geral, esta abordagem pode ser vista como um algoritmo comportamental distribuído de busca multidimensional onde cada partícula do enxame é candidata a solução do problema e possui uma velocidade atual, uma posição atual e o conhecimento da sua melhor posição pessoal. Dessa forma, cada partícula i do enxame, em um instante t , pode ser representada através dos seguintes vetores:

- Sua posição atual em um espaço de busca D -dimensional: $\vec{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$;
- A melhor posição que ela encontrou até então: $\vec{p}_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{iD}(t))$, também conhecida como $pBest$;
- A sua velocidade atual: $\vec{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$.

As partículas também guardam um vetor, $\vec{p}_{ig}(t) = (p_{ig1}(t), p_{ig2}(t), \dots, p_{igD}(t))$, conhecido como $gBest$, que representa o seu conhecimento social, ou seja, esse vetor armazena a melhor posição da melhor partícula de todo o enxame.

Inicialmente, a posição e a velocidade de cada partícula são definidas aleatoriamente e, durante a execução do algoritmo, as partículas movem-se no espaço de busca para encontrar a melhor solução possível. A cada iteração, os vetores de posição e de velocidade de cada partícula são atualizados através das seguintes equações:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + c_1\epsilon_1[\vec{p}_i(t) - \vec{x}_i(t)] + c_2\epsilon_2[\vec{p}_{ig}(t) - \vec{x}_i(t)] \quad (3.8)$$

e

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1), \quad (3.9)$$

onde $i = 1, \dots, N$, c_1 e c_2 são os parâmetros cognitivo e social, respectivamente, com o cognitivo representando a influência do conhecimento individual e, o social, a influência do conhecimento coletivo; ϵ_1 e ϵ_2 são dois números aleatórios gerados por uma distribuição de probabilidade uniforme dentro do intervalo $[0, 1]$.

As forças, que influenciam o movimento de cada partícula no espaço de busca e resultam na atualização de sua posição, podem ser representadas matematicamente por vetores, definidos da seguinte forma:

- **Vetor inércia:** representa o movimento atual da partícula, ou seja, a velocidade corrente que impulsiona a partícula para a região onde ela aposta que está a solução. Na Equação (3.8), esse vetor corresponde a $\vec{v}_i(t)$;
- **Vetor memória:** corresponde à componente cognitiva da partícula, ou seja, é a relação entre a posição atual e a melhor posição encontrada por aquela partícula. Na Equação (3.8), esse vetor é representado pelo termo $c_1\epsilon_1(\vec{p}_i(t) - \vec{x}_i(t))$;
- **Vetor cooperação:** representa a influência do enxame em uma determinada partícula. É a relação entre a melhor posição encontrada pelo enxame e a melhor posição encontrada pela partícula. Na Equação (3.8), o vetor é representada pelo termo $c_2\epsilon_2(\vec{p}_{ig}(t) - \vec{x}_i(t))$.

A atuação desses vetores pode ser observada pelo esquema gráfico da Figura 5, no qual a posição atual da partícula é $\vec{x}_i(t)$ e a posição da partícula após o processo de atualização é $\vec{x}_i(t+1)$.

Além das partículas terem sua velocidade e posição atualizadas a cada iteração, os valores do seu $pBest$ e $gBest$ também são verificados. Se a nova posição da partícula for melhor

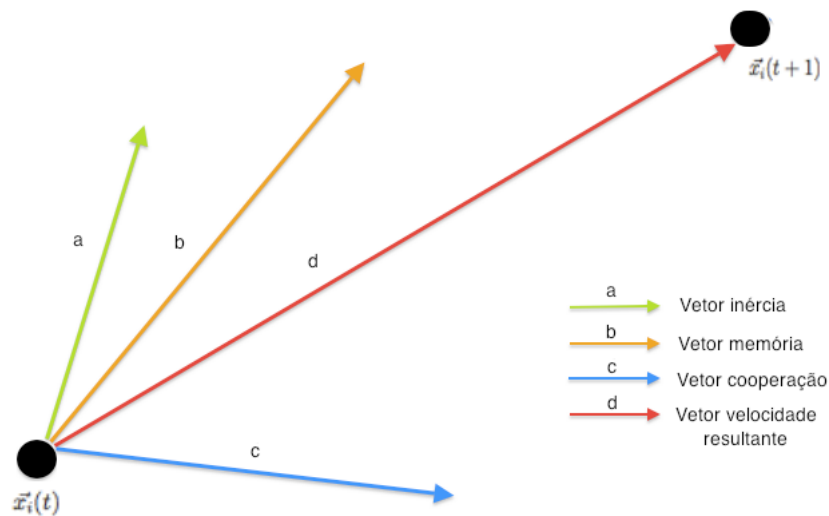


Figura 5: Vetores que representam as forças que influenciam o movimento de uma partícula.

que seu atual $pBest$, ele será atualizado. O mesmo acontece com o $gBest$: se for encontrada uma nova melhor posição para a melhor partícula do enxame ou uma nova melhor partícula, o $gBest$ é atualizado.

O algoritmo do PSO tem sua execução interrompida quando uma condição de parada determinada é atingida. Essa condição pode ser o número de iterações ou um limiar de aceitação que indica se o enxame chegou a um ponto cujo desempenho não melhore significativamente, ou seja, o enxame convergiu para um ponto sub-ótimo do problema. Esse desempenho é calculado pra cada partícula, a cada iteração, e é chamado de *fitness*. O Algoritmo 3 ilustra o funcionamento do PSO.

Algoritmo 3: Pseudocódigo do PSO.

Entrada: *numeroParticulas*, *numeroDimensoes* e *numeroIteracoes*

```

1 início
2   inicializaEnxame()
3   para i = 1 até numeroIteracoes faça
4     para j = 1 até numeroParticulas faça
5       para k = 1 até numeroDimensoes faça
6         atualizaVelocidade();
7         atualizaPosicao();
8       calculaFitness();
9       atualizaPBest();
10      atualizaGBest();
11 fim
  
```

3.3 PSO Aplicado a Otimização Multiobjetiva

A técnica de Otimização por Enxame de Partículas é muito utilizada e obtém excelentes resultados para uma grande variedade de problemas de busca e de otimização que possuem um único objetivo, uma vez que apenas uma solução ótima é encontrada. Como visto anteriormente, em problemas de otimização multiobjetiva, faz-se necessário encontrar um conjunto de soluções ótimas que, muitas vezes, buscam satisfazer necessidades conflitantes. Apesar desse fato, a PSO é uma candidata natural para solução de problemas com mais de um objetivo devido a sua simplicidade de implementação, seu bom desempenho e por ser uma técnica baseada no conceito de população [Reyes-Sierra e Coello 2006].

Entretanto, quando se pensa em aplicar a PSO à otimização multiobjetiva, fica evidente, com o que foi apresentado na Seção 3.2, que modificações precisam ser realizadas ao algoritmo padrão da PSO. Para realizar essas modificações, três questões precisam ser consideradas:

1. Como selecionar partículas de maneira que soluções não dominadas tenham preferência sobre as dominadas?
2. Como guardar todas soluções não dominadas encontradas ao longo da execução do algoritmo para que, ao final, seja possível retornar as soluções não dominadas de enxames passados e não apenas as encontradas no enxame atual?
3. Como garantir diversidade ao enxame para evitar que ele convirja para uma única solução?

Diante dessas questões, diversas foram as propostas sugeridas para estender a capacidade da PSO para otimizar problemas multiobjetivo. A seguir, algumas dessas propostas serão apresentadas.

MOPSO

Proposto por [Coello, Pulido e Lechuga 2004], este algoritmo baseia-se na ideia das partículas armazenarem suas experiências, após cada iteração, em um arquivo externo. O espaço de objetivos, onde a busca acontece, é dividido em hipercubos, onde cada hipercubo terá um valor de *fitness* que depende do número de partículas nele inseridas.

A escolha das melhores soluções no enxame, ou seja, os líderes sociais de cada partícula, é realizada usando roleta. Nessa técnica, seleciona-se um hipercubo e escolhe-se aleatoriamente

uma das partículas inseridas nele para ser líder. Esta abordagem também utiliza um operador de mutação para aumentar a habilidade exploratória das partículas e para evitar a convergência para um *Pareto Front* falso.

MOPSO-CDLS

Baseado na abordagem de [Raquel e Naval 2005], esta proposta de [Tsou et al. 2007] usa o conceito de *Crowding Distance*, CD, que analisa a distância entre as partículas e é importante para saber como as soluções estão distribuídas e na escolha dos líderes sociais.

A escolha do líder social de cada partícula, o $gBest$, é feita considerando duas situações: se a partícula é dominada pelas soluções que são as 10% com menor CD, o líder é escolhido aleatoriamente entre essas soluções; caso contrário, o líder também é escolhido aleatoriamente, mas entre todo o arquivo externo.

O líder cognitivo de cada partícula, seu $pBest$, é atualizado se a nova posição domina o líder cognitivo atual. No caso das soluções serem incomparáveis, esse líder é escolhido de forma aleatória entre essas duas opções. Como uma maneira de melhorar as habilidades de exploração e a velocidade de convergência, esta abordagem faz uso de um mecanismo de busca local no arquivo externo.

MOPSO-CDR

Nessa proposta de [Santana 2009] baseada na abordagem de [Tsou et al. 2007], a escolha do líder social, ou $gBest$, é feita usando *Crowding Distance* na roleta e esses conceitos ainda servem para prevenir um número excessivo de soluções não dominadas no arquivo externo. Ao final cada iteração, o arquivo externo é ordenado pelo CD e cada partícula terá seu líder social escolhido através da aplicação da roleta, com soluções em regiões menos populosas tendo mais chances de serem escolhidas.

Quanto à atualização do líder cognitivo, ou $pBest$, o MOPSO-CDR apresenta uma abordagem completamente nova. Se a nova posição da partícula dominar seu atual $pBest$, ele é então atualizado. No caso da nova posição ser incomparável com o atual $pBest$, o algoritmo procura no arquivo externo por soluções com distância Euclidiana mínima tanto para o $pBest$ quanto para a nova posição. Se a solução mais próxima da nova posição estiver em uma região menos populosa do que a solução mais próxima do $pBest$, a nova posição atualizará o $pBest$; caso contrário, o seu valor atual permanece.

Assim como no MOPSO [Coello, Pulido e Lechuga 2004], esta proposta também utiliza um operador de mutação. Esse operador, inicialmente, afeta todas as partículas, mas, ao longo da execução do algoritmo, o número de partículas afetadas diminui.

Pelos resultados obtidos com essa nova proposta de aplicação da PSO a problemas multi-objetivos, ela foi escolhida para ser utilizada no desenvolvimento deste trabalho de conclusão de curso. O Algoritmo 4 detalha o funcionamento do MOPSO-CDR.

Algoritmo 4: Pseudocódigo do MOPSO-CDR.

Entrada: *numeroParticulas*, *numeroDimensoes* e *numeroIteracoes*

Saída: arquivo externo

```
1 início
2   inicializaEnxame();
3   inicializaLideresNoArquivoExterno();
4   qualificaLideresPorCD();
5   i = 1;
6   enquanto i = 1 ≤ numeroIteracoes faça
7     para j = 1 até numeroParticulas faça
8       aplicaMutacao();
9       selecionaLider();
10      para k = 1 até numeroDimensoes faça
11        atualizaVelocidade();
12        atualizaPosicao();
13      calculaFitness();
14      atualizaPBest();
15    atualizaLideresNoArquivoExterno();
16    qualificaLideresPorCD();
17    i = i + 1;
18  escreveResultadosNoArquivoExterno();
19 fim
```

4 *Proposta e Arranjo Experimental*

Diante da importância cada vez maior das telecomunicações e do importante papel que as redes *ad hoc* podem desempenhar nesse cenário, o objetivo deste trabalho é propor uma forma de otimizar essas redes, visando dois aspectos críticos: vazão e atraso médio. Considerando que, devido às características inerentes às redes *ad hoc*, o roteamento nessas redes não é trivial e exerce influência direta no desempenho da rede como um todo, a proposta do trabalho é buscar otimizar essas redes a partir de seu roteamento. Muitos são os protocolos de roteamento para redes *ad hoc*, como visto na Seção 2.1, sendo o AODV o mais abordado no estado-da-arte da literatura especializada, o que resultou na sua escolha como protocolo a ser modificado para a realização da otimização sugerida neste trabalho.

Tomando como base a proposta do trabalho de [Pereira 2010], o algoritmo do AODV foi modificado no que diz respeito à recuperação de quebras de enlace em rotas ativas. Em [Pereira 2010], além do número de saltos percorridos pelo pacote até o nó predecessor à quebra e do número de saltos que separam esse nó do destino, parâmetros considerados na implementação padrão do AODV [Perkins, Belding-Royer e Das 2003], a conectividade tanto do nó fonte quanto do predecessor à quebra passaram também a ser consideradas. Juntamente com essa modificação, pesos foram adicionados a cada um dos parâmetros utilizados para escolher o mecanismo de recuperação mais adequado. Como o objetivo era otimização do atraso médio da rede, esses pesos foram definidos através da técnica de Inteligência Computacional, PSO. Porém, como mostrado na Seção 3.3, a PSO só é aplicável na resolução de problemas de otimização de único objetivo, fazendo com que, para a proposta deste trabalho de conclusão de curso, uma abordagem multiobjetiva da PSO fosse utilizada, especificamente o algoritmo MOPSO-CDR, uma vez que o objetivo é a otimização simultânea da vazão e do atraso médio.

A fim de verificar como a otimização de duas métricas específicas pode afetar outros aspectos da rede, a sua sobrecarga e o número médio dos saltos percorridos pelos pacotes também foram analisadas. A seguir, as quatro métricas analisadas são descritas:

- **Vazão:** a razão entre os pacotes de dados recebidos no nós destinos e os pacotes de

dados enviados pelos nós fontes;

- **Atraso médio:** inclui todos os possíveis atrasos causados desde o momento em que os pacotes de dado saem dos nós fontes até quando chegam ao destino desejado;
- **Sobrecarga:** o total de pacotes de controle transmitidos na rede;
- **Número médio de saltos:** número médio de saltos por pacote entregue entre todos os pares origem-destino.

4.1 AODV Utilizando o MOPSO-CDR

O MOPSO-CDR foi utilizado para encontrar os melhores valores para os pesos A, B, C e D associados aos parâmetros que determinam qual o mecanismo de recuperação de quebra de enlace deve ser utilizado, de forma a contribuir na otimização da vazão e do atraso médio da rede.

As partículas do enxame percorrem um espaço de busca com quatro dimensões para encontrar as soluções que possam satisfazer os dois objetivos considerados, da melhor forma e ao mesmo tempo. A cada iteração, depois de ter sua posição atualizada, cada partícula chama a função de cálculo de *fitness* e, ao chamar essa função, ela tem os valores do vetor que armazena a sua posição atual escritos em um arquivo. Todas as partículas escrevem no mesmo arquivo e a última linha do arquivo sempre guarda a posição mais atual da partícula que está tendo seu *fitness* calculado no momento. Dessa forma, os valores do vetor da posição podem ser usados como os pesos da decisão do mecanismo de recuperação. A Figura 6 mostra um trecho de um exemplo desse arquivo.

```
-0.661817 -0.758494 0.57576 -0.287538  
0.199686 -0.378986 -0.557728 0.933419  
0.761494 -0.128798 0.777079 -0.307146  
0.949018 -0.770871 -0.194986 0.900105  
-0.743653 -0.503449 -0.494121 0.626908  
0.323235 0.779955 0.824467 0.578443
```

Figura 6: Exemplo do arquivo que guarda os valores do vetor de posição das partículas.

Uma vez que a posição da partícula foi guardada em arquivo, ainda no cálculo do *fitness*, executa-se a simulação de uma rede *ad hoc* que usa o AODV como seu protocolo de roteamento. Essa simulação, quando precisa fazer a reparação de uma rota, lê os valores da posição da partícula do arquivo e os utiliza como os pesos dos parâmetros de decisão. Ao término da simulação, o resultado do atraso médio e o da vazão são escritos em um outro arquivo para que a função de *fitness* possa ter acesso a eles e, assim, atualize a medida de desempenho

da partícula. Da mesma forma que acontece com o arquivo que armazena os valores do vetor posição das partículas, esse arquivo guarda o valor do *fitness* de todas as partículas ao longo da simulação, sendo a última linha correspondente à medida de desempenho da partícula para qual o cálculo está sendo realizado no momento. Um trecho de um exemplo desse arquivo é mostrado pela Figura 7, com a primeira coluna sendo os valores do atraso médio e a segunda os da vazão.

```
0.0262248 0.970037
0.11037 0.994893
0.0658186 0.983916
0.136329 0.994329
0.136329 0.994329
0.111438 0.991907
0.136329 0.994329
```

Figura 7: Exemplo do arquivo que guarda o valor do *fitness* das partículas.

O esquema do funcionamento conjunto do AODV e do MOPSO-CDR é mostrado pelo diagrama da Figura 8.

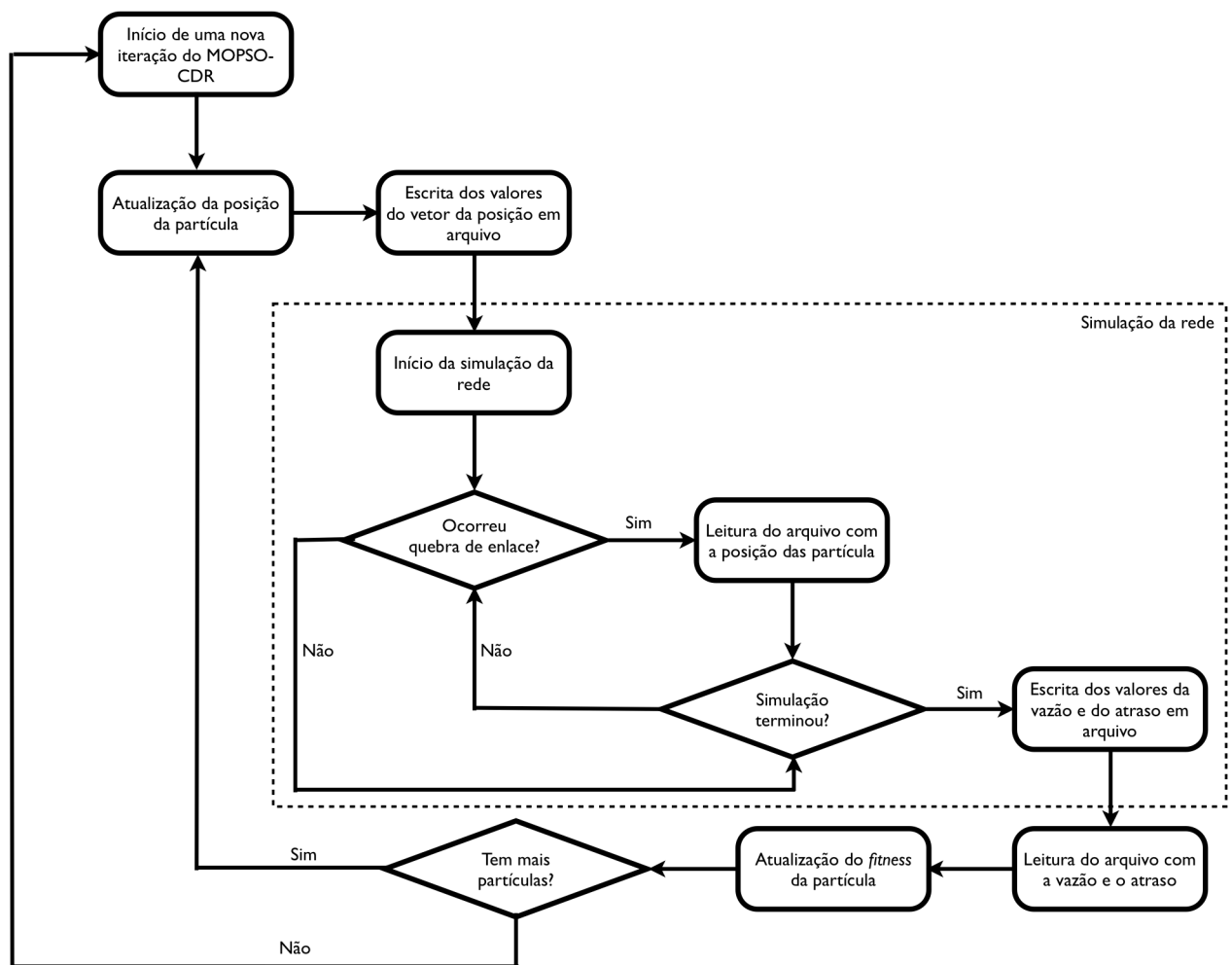


Figura 8: Diagrama do funcionamento conjunto do AODV e MOPSO-CDR.

4.2 Arranjo Experimental

O arranjo experimental desenvolvido para a proposta apresentada consistiu inicialmente na implementação de uma nova função de *fitness* para o código do MOPSO-CDR de [Santana 2009]. Essa nova função refletiu o que foi descrito na Seção 4.1 para que o MOPSO-CDR fosse integrado ao AODV.

No que diz respeito ao AODV, todas as modificações realizadas por [Pereira 2010] e todas as necessárias para que ele utilizasse o MOPSO-CDR foram feitas na implementação do seu código encontrada no *Network Simulator*.

O *Network Simulator*, ou ns-2 [The Network Simulator - ns-2], em referência a sua geração atual, é um simulador de eventos discretos desenvolvido para a pesquisa com Redes de Telecomunicações. O ns-2 é extremamente popular devido a sua extensibilidade por ser um *software* de código aberto e também porque possui um grande suporte para diversos protocolos de redes e pode ser usado na simulação tanto de redes cabeadas quanto sem fio. No meio acadêmico, ele é bastante usado para simulações de protocolos de roteamento e de redes *ad hoc*, sendo, portanto, o simulador escolhido para a realização deste trabalho.

Na implementação padrão do AODV no ns-2, o monitoramento da conectividade entre os nós participantes de uma rota ativa é realizado através das informações obtidas a partir da camada de enlace. Entretanto, como o uso de mensagens *hello* torna a obtenção das informações sobre a conectividade dos nós mais simples, a implementação do AODV no ns-2 foi ajustada para fazer uso delas em vez das informações da camada de enlace.

A função de *fitness* do MOPSO-CDR, para obter os valores da medida de desempenho de uma partícula, chama o ns-2 que, por sua vez, simula um cenário de uma rede *ad hoc*. Os cenários construídos para as simulações aproximam-se ao máximo dos cenários descritos em trabalhos relevantes sobre o AODV [Das, Perkins e Royer 2000], [Broch et al. 1998] e possuem as seguintes configurações: rede com área de 1500 x 300 m, 50 nós, 10 e 30 conexões fim-a-fim ¹, velocidade máxima de movimento dos nós 20 m/s ², tempo de pausa variando entre 0, 30, 60, 120, 300, 600 e 900 ³ segundos e tempo total de simulação de 900 segundos.

Vale notar que, diferentemente de [Pereira 2010], neste trabalho de conclusão de curso, procurou-se também analisar a aplicação de Computação Inteligente na otimização de re-

¹Conexões definidas aleatoriamente pelo *cbrgen.tcl* [Wireless Simulation Scenarios in ns-2] em sua versão modificada por [Filho 2008].

²As posições dos nós ao longo da simulação foram definidas aleatoriamente pelo *setdest* [Wireless Simulation Scenarios in ns-2].

³O tempo de pausa estipula o grau de mobilidade da rede, com tempo de pausa 0 significando mobilidade total do nós e 900, nenhuma mobilidade.

des *ad hoc* para cenários com 10 conexões fim-a-fim. Outras configurações desses cenários que merecem ser citadas são as seguintes: nós com raio de alcance de 250 m, camadas física e MAC no padrão IEEE 802.11, modelo de mobilidade *Random Waypoint* (RWP) [BETTSTETTER e WAGNER], modelo de propagação *Two-Ray Ground* [FALL 2009], padrão de tráfego UDP do tipo CBR para as conexões, taxa de envio de pacotes de 4 pacotes/segundo e pacotes com 512 *bytes* de tamanho.

Uma vez que o ns-2 simula o cenário de rede, um arquivo chamado *trace* é gerado, contendo informações detalhadas sobre o comportamento da rede durante todo o tempo de simulação. Aplicando *scripts* de extração de dados apropriados para esse arquivo, é possível obter os valores de diversas métricas, incluindo a vazão e o atraso médio. Entretanto, pelas configurações dos cenários sendo considerados e pelo nível de detalhamento dos arquivos de *trace*, eles podem ser demasiadamente grandes ⁴, tornando consideravelmente custoso, em termos de tempo, a extração de métricas a partir deles com o uso *scripts* de leitura de arquivo. Para solucionar esse problema, o código fonte do ns-2 foi modificado para, além de refletir as mudanças no algoritmo do AODV, armazenar, durante a própria simulação, os valores das métricas a serem otimizadas pelo MOPSO-CDR, evitando a dependência do *trace*.

Para cada tempo de pausa sendo considerado nos cenários de rede, O MOPSO-CDR foi executado por 300 iterações considerando um enxame de 20 partículas em um espaço de busca de 4 dimensões no intervalo $[-1, 1]$. Ao final das iterações, o arquivo externo é gerado e nele estão armazenadas as melhores soluções não dominadas encontradas, ou seja, o *Pareto Front*, para as iterações 0, 100, 200 e 300. A Figura 9 mostra um exemplo do formato do arquivo externo gerado, onde a primeira linha depois da identificação da partícula diz a posição dela na iteração em que ela foi armazenada e, logo embaixo, o *fitness* associado.

```
Particula 11:  
0.951087, -0.972783, 0.691538, -0.394226  
0.100013, 0.991744  
  
Particula 14:  
-0.668585, 0.489905, -0.708175, -0.85788  
0.0241479, 0.970061  
  
Particula 17:  
0.983776, -0.364892, 0.686614, -0.339735  
0.0290488, 0.978969  
  
Particula 12:  
1, -0.217855, 0.86443, -0.201802  
0.0300717, 0.980627
```

Figura 9: Exemplo de arquivo externo gerado pelo MOPSO-CDR.

Uma vez que as melhores soluções para o problema multiobjetivo foram encontradas,

⁴No caso dos cenários com as configurações descritas aqui, um arquivo de *trace* pode ter até 750 MB.

os *Pareto Front* são analisados e três pontos de cada são escolhidos: o que representa uma solução com o menor atraso, o que representa uma com a maior vazão e um que seja equilibrado entre os dois objetivos. Feito isso, os valores do vetor de posição de cada um desses pontos serão utilizados para simulação de 10 cenários de rede diferentes para cada tempo de pausa sendo analisado. Quando a simulação dos cenários de um tempo de pausa é concluída, é possível saber o valor médio da vazão e do atraso médio. Quanto a extração das métricas sobrecarga e número médio de saltos, o *trace* é utilizado, uma vez que essas são métricas secundárias e também mais simples de serem extraídas a partir de um *script*, o que torna essa abordagem mais conveniente do que a modificação do código fonte do ns-2 para a extração automática.

Todos os código de algoritmos e de simulações foram executadas no *grid* computacional do curso de Engenharia de Computação da Escola Politécnica da Universidade de Pernambuco. O *grid* é composto por 12 máquinas cujas configurações são um processador Intel® Core™2 Quad Q6600 de 2,40 GHz ou Q8200 2,33 GHz com 4 GB de memória RAM DDR3.

5 *Resultados e Discussão*

Uma vez realizada a execução conjunta do AODV e do MOPSO-CDR, o arquivo externo gerado armazena as melhores soluções para o problema de otimização da vazão e do atraso médio em redes *ad hoc* que usam o AODV como seu protocolo de roteamento, considerando as modificações descritas na Seção 4.1.

O arquivo externo armazena a posição e o *fitness* associado àquela posição das melhores soluções, ou seja, as soluções pertencentes ao *Pareto Front*, encontradas nas iterações 0, 100, 200 e 300 (a última). A partir disso, para cada tempo de pausa, gráficos foram gerados com os *Pareto Front* de cada iteração citada, em função dos valores de *fitness*, para que fosse possível analisar qual a melhor solução para o atraso médio, para a vazão e qual a mais equilibrada para essas duas métricas.

Uma vez escolhidas as melhores soluções, as posições a elas associadas foram utilizadas como pesos dos parâmetros de escolha do mecanismo de reparação utilizado quando uma quebra de enlace acontece. Para cada tempo de pausa, 10 cenários diferentes de uma rede *ad hoc* foram simulados levando em conta cada uma das soluções encontradas: melhor vazão, melhor atraso e equilibrada. A média dos resultados geraram gráficos que comparam a evolução das métricas analisadas, incluindo sobrecarga e número médio de saltos, para cada melhor solução encontrada, à medida que o tempo de pausa aumentam.

Por fim, a solução que apresentou os melhores resultados de uma maneira geral entre todas as métricas foi escolhida para ter seus resultados comparados aos do AODV e aos do AODV com PSO de [Pereira 2010].

Nas seções deste capítulo, os resultados descritos são apresentados e uma breve discussão acerca deles é realizada.

5.1 *Análise dos Paretos Front*

Para cenários com 10 conexões fim-a-fim, a Figura 10 apresenta os *Paretos Front* das iterações 0, 100, 200 e 300.

Pode-se observar que, à medida que mais iterações são realizadas, melhores e mais diversas são as soluções encontradas. Entretanto, a partir de 100 iterações, muitos dos resultados obtidos são os mesmos dos encontrados para 200 e 300 iterações, além da distância das soluções ser pequena.

Diferentemente do que é observado em geral, para um tempo de pausa de 900 segundos, pouca é a diversidade das soluções encontradas. Isso deve-se ao fato de que, como não há a mobilidade dos nós, não há quebras de enlaces, já que o tempo de pausa é igual ao tempo total de duração da simulação.

Já em cenários com 30 conexões fim-a-fim, como mostrado na Figura 11, o número maior de conexões implica, de modo geral, em uma carga muito maior na rede do que para cenários com 10 conexões. Nesse caso, a diversidade de soluções encontradas é bem menor do que para cenários com 10 conexões e somente observada em cenários com tempos de pausa intermediários (tempos de pausa 60, 120 e 300), ou seja, quanto maior o tráfego na rede e em casos extremos de mobilidade, mais difícil é a otimização através do uso de enxame de partículas. Entretanto, a dificuldade de otimização para cenários com tempos de pausa 0 e 30 deve-se ao fato desses cenários serem bem caóticos, com muito tráfego e muito movimento; enquanto que para tempo de pausa 900 deve-se ao fato de não haver quebras de enlaces, uma vez que os nós não se movem durante todo o período da simulação.

Entretanto, com exceção ao tempo de pausa 300, é possível notar um comportamento semelhante aos cenários com 10 conexões: muitas das soluções encontradas para 100, 200 e 300 iterações são similares.

5.1 Análise dos Pareto Front

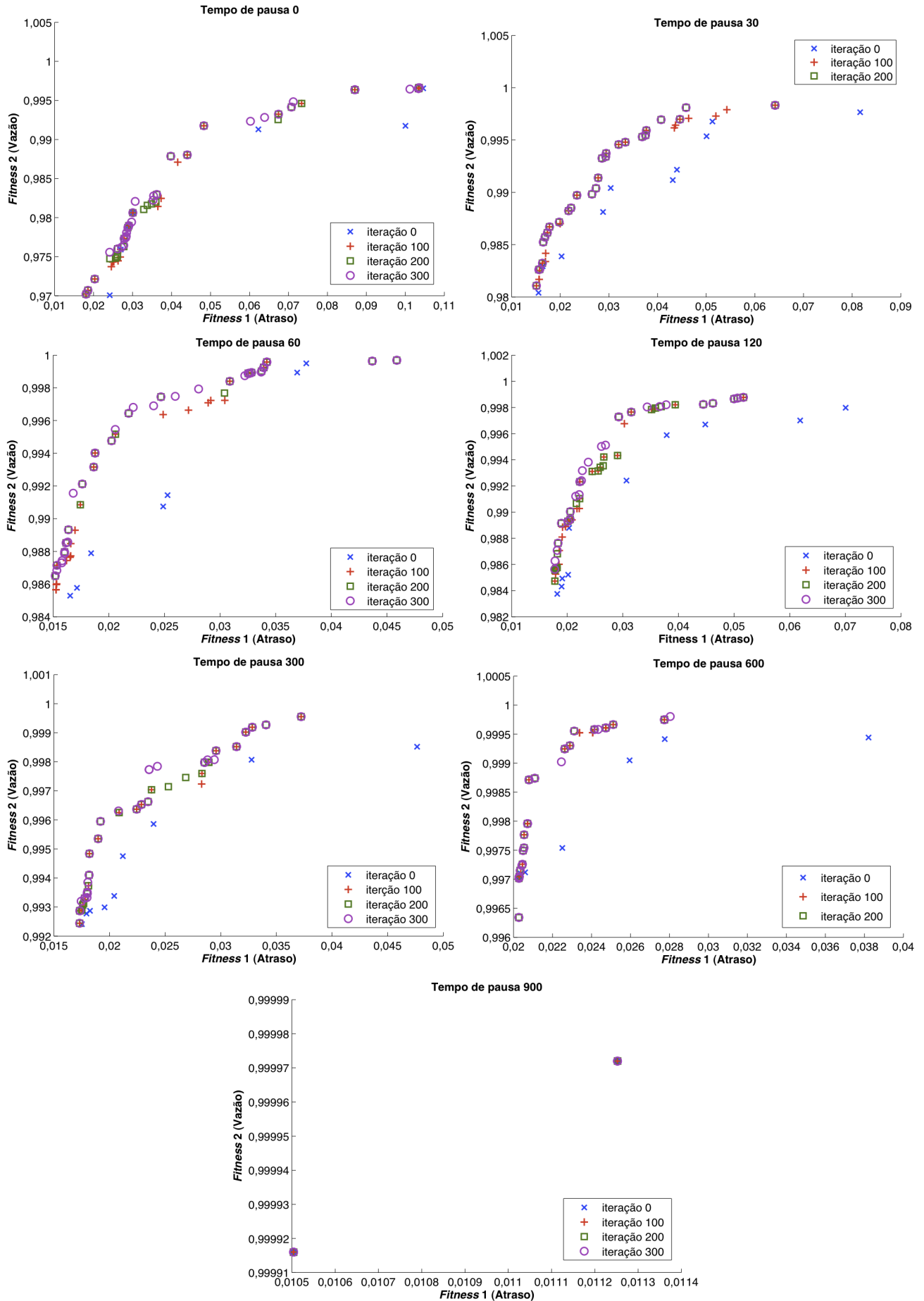


Figura 10: Pareto Front dos cenários com área de 1500 x 300 metros e 10 conexões para as iterações 0, 100, 200 e 300.

5.1 Análise dos Pareto's Front

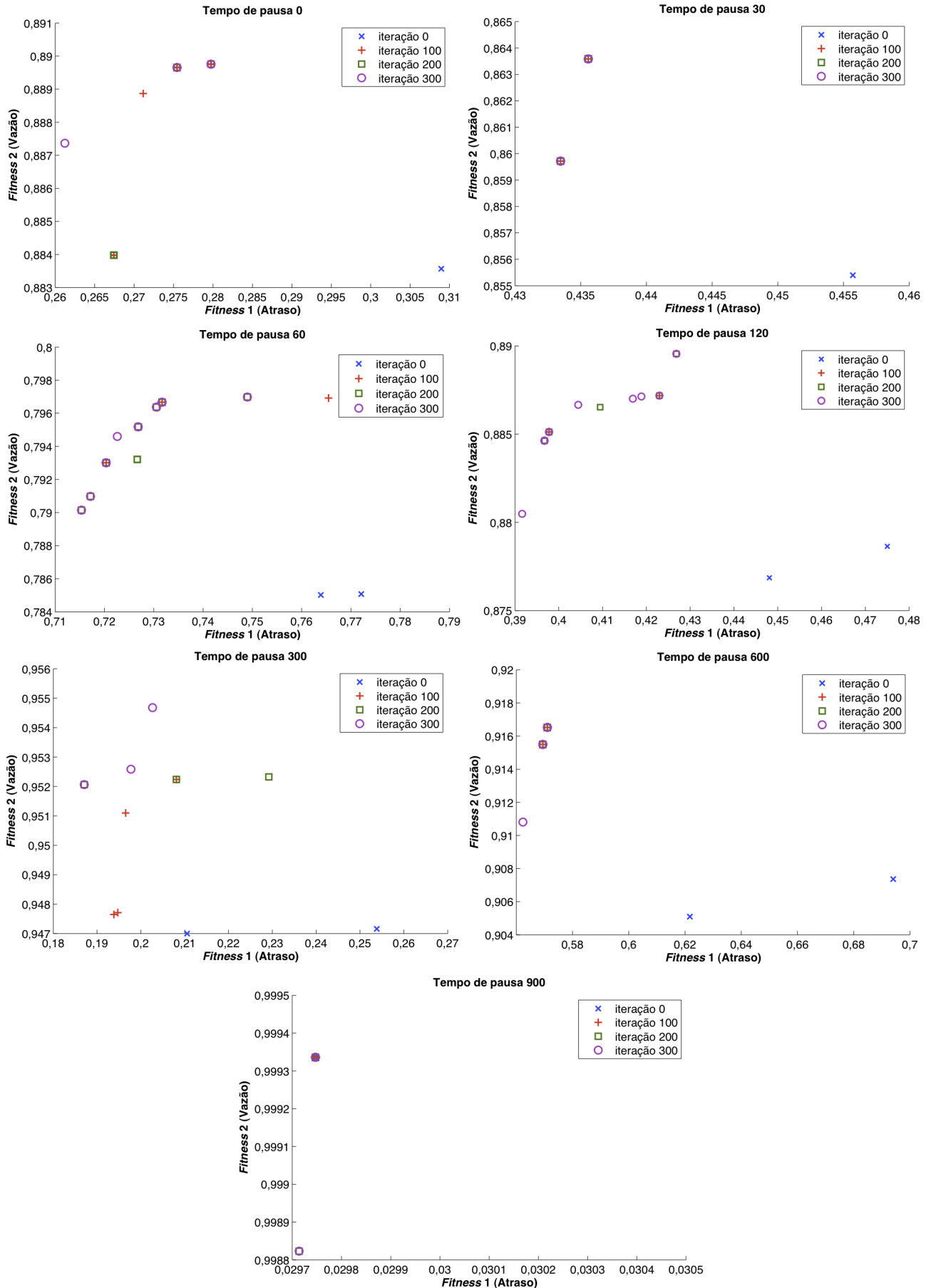


Figura 11: Pareto's Front dos cenários com área de 1500 x 300 metros e 30 conexões para as iterações 0, 100, 200 e 300.

Depois dos *Paretos Front* para 10 e 30 conexões serem analisados, as melhores soluções em termos de atraso, de vazão e equilibradas foram escolhidas. As tabelas 1 e 2 mostram quais são essas soluções para 10 e 30 conexões, respectivamente. Também são mostradas as posições atreladas a essas soluções e que foram usadas como pesos na escolha do mecanismo de reparação nas simulações cujos resultados serão vistos nas próximas seções.

Tabela 1: Tabela com os parâmetros escolhidos como melhores a partir da análise dos *Paretos Front* para cenários com 10 conexões.

10 conexões fim-a-fim			
Tempo de pausa (s)	Tipo da solução	<i>Fitness</i>	Posição
0	melhor atraso	0,0180706; 0,970207	-0,419888; 1; 0,800697; 0,141803
	equilibrada	0,0482861; 0,991758	-0,47601; -1; -1; -0,615267
	melhor vazão	0,103525; 0,996612	0,96195; -0,386335; 0,776349; 1
30	melhor atraso	0,0150995; 0,981074	0,801543; 0,00474826; 1; -0,353514
	equilibrada	0,037694; 0,995916	1; -1; 1; -0,648458
	melhor vazão	0,0458631; 0,9981	0,601631; -0,263897; 0,0527579; 0,371799
60	melhor atraso	0,0151882; 0,986506	-0,561251; 0,331244; 1; -0,156173
	equilibrada	0,0217852; 0,996442	-0,726327; 0,230398; -0,0726966; 0,220323
	melhor vazão	0,0458831; 0,999692	-0,678507; -0,608381; -0,433803; 0,059578
120	melhor atraso	0,0177494; 0,985636	0,843615; -0,509697; 0,994817; -0,988245
	equilibrada	0,0315182; 0,997651	1; -0,574238; -0,480543; 0,342303
	melhor vazão	0,0516719; 0,99877	0,548325; -1; 0,0265042; 0,0460785
300	melhor atraso	0,0173165; 0,992441	0,25974; -0,39564; -1; -0,475078
	equilibrada	0,0235574; 0,997733	0,990369; -0,678235; 1; -0,32506
	melhor vazão	0,0372233; 0,999552	0,415589; -0,3946; 0,784249; 0,0523165
600	melhor atraso	0,0202708; 0,996343	-0,931941; -0,387844; 0,416687; -0,851029
	equilibrada	0,0231142; 0,999553	0,65801; 0,399458; -1; 1
	melhor vazão	0,0280412; 0,999804	-0,934814; 0,0329507; -0,053208; -0,00163037
900	melhor atraso	0,0105037; 0,999916	0,258746; 0,270222; -0,0968392; -0,706767
	equilibrada	0,0105037; 0,999916	0,175268; -0,0593986; 0,264919; -0,136278
	melhor vazão	0,0112523; 0,999972	-0,273719; -0,287985; -0,418342; 0,0779998

Tabela 2: Tabela com os parâmetros escolhidos como melhores a partir da análise dos *Pareto Front* para cenários com 30 conexões.

30 conexões fim-a-fim			
Tempo de pausa (s)	Tipo da solução	<i>Fitness</i>	Posição
0	melhor atraso	0,261218; 0,887364	0,21863; 0,418727; -0,984026; 0,100802
	equilibrada	0,271153; 0,888868	0,672819; 1; -0,264331; 0,122415
	melhor vazão	0,27975; 0,889757	0,709135; 0,387624; -0,996733; 0,137936
30	melhor atraso	0,433462; 0,859719	0,824866; 0,116376; 0,611592; -0,655593
	equilibrada	0,435571; 0,863584	0,779411; 0,0726807; 0,727896; -0,799322
	melhor vazão	0,435571; 0,863584	0,725704; 0,076439; 0,70539; -0,778777
60	melhor atraso	0,715333; 0,790145	0,261033; 0,149301; 0,870451; -1
	equilibrada	0,730549; 0,796371	1; 0,219857; 0,883059; -0,595133
	melhor vazão	0,748994; 0,796981	0,824947; 0,175603; 0,923633; -0,803324
120	melhor atraso	0,391641; 0,880482	-0,0130239; -0,13404; -0,850713; -0,539825
	equilibrada	0,404429; 0,886657	0,012051; -0,201471; -0,745108; -0,452191
	melhor vazão	0,42686; 0,889548	0,00374719; -0,204913; -0,769461; -0,425143
300	melhor atraso	0,187073; 0,952066	0,838416; 0,00568799; 0,165008; 0,0249806
	equilibrada	0,197737; 0,952588	0,807765; -0,106905; 0,231296; -0,121855
	melhor vazão	0,202667; 0,954679	0,757796; 0,0561122; 0,509929; -0,0498297
600	melhor atraso	0,562361; 0,910807	1; -0,498177; -1; -0,537496
	equilibrada	0,571023; 0,916525	0,996193; -0,718873; -0,80669; -0,724046
	melhor vazão	0,571023; 0,916525	0,998582; -0,716821; -0,827784; -0,710775
900	melhor atraso	0,029714; 0,998823	0,897475; 0,318643; 0,306551; 0,521992
	equilibrada	0,0297472; 0,999336	0,700004; -1; 0,0455408; 0,760603
	melhor vazão	0,0304053; 0,999355	0,256785; -0,733174; -0,0123804; 0,64273

5.2 Análise dos resultados do AODV utilizando o MOPSO-CDR

Observando os resultados das simulações executadas considerando cada uma das soluções mostradas na Tabela 1, pode-se ver na Figura 12 que, para cenários com 10 fontes, os resultados para as soluções de melhor atraso foram melhores para todas as métricas, com exceção da vazão. Entretanto, como a diferença do resultado da vazão para melhor atraso é pequena quando comparado aos resultados para melhor vazão e equilibrado e diante dos resultados das outras métricas, os resultados para melhor atraso foram os escolhidos para comparação com o AODV original e o AODV usando PSO.

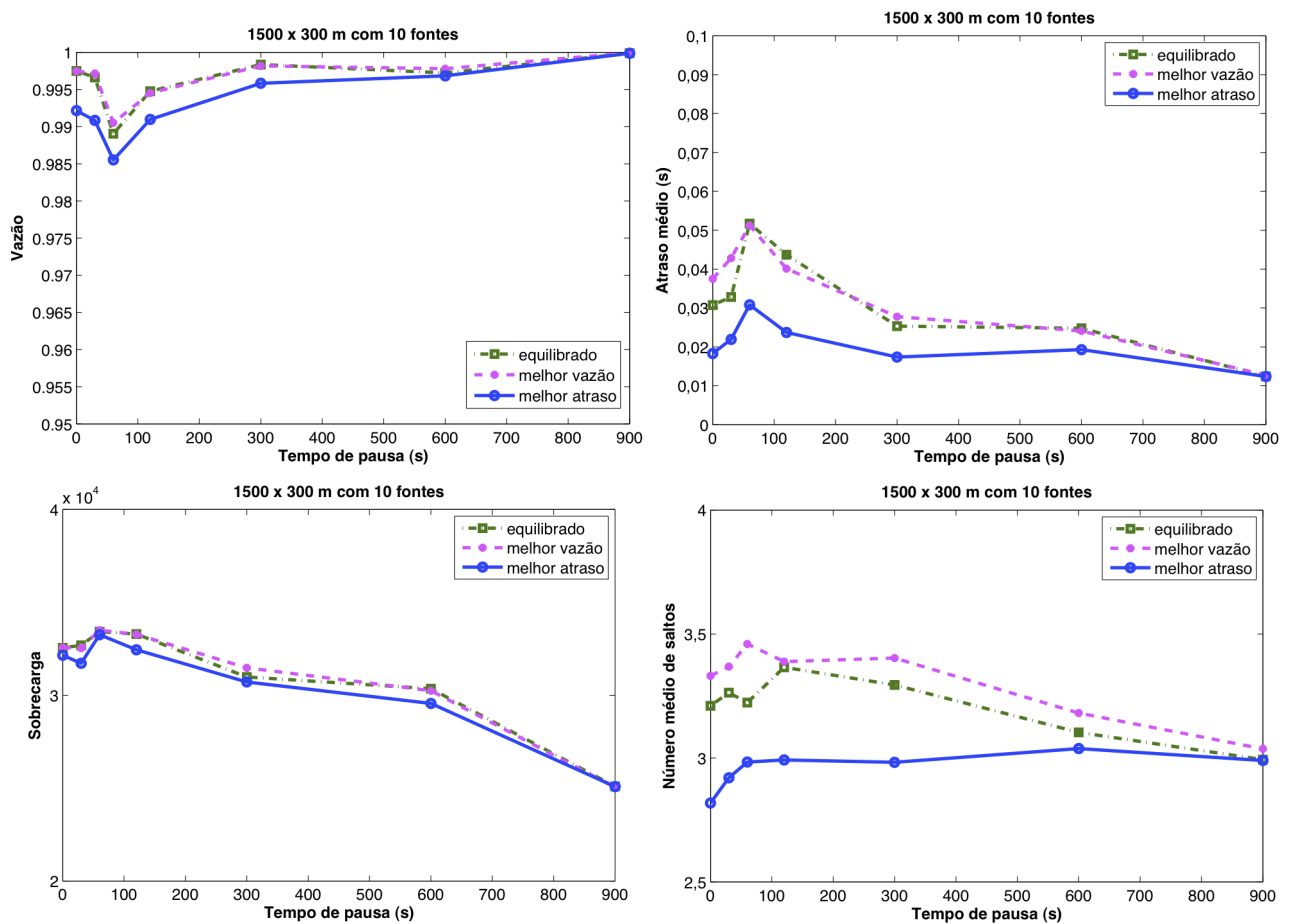


Figura 12: Gráficos comparativos dos resultados das simulações para cenários com 10 conexões considerando as melhores soluções do MOPSO-CDR para atraso e vazão e a mais equilibrada.

Para 30 conexões, a Figura 13 mostra que há um maior equilíbrio entre as soluções, com os resultados para melhor atraso sendo melhores para tempos de pausa 900. Portanto, para esse caso, os resultados para melhor atraso também foram os escolhidos para comparação com o AODV original e o AODV usando PSO.

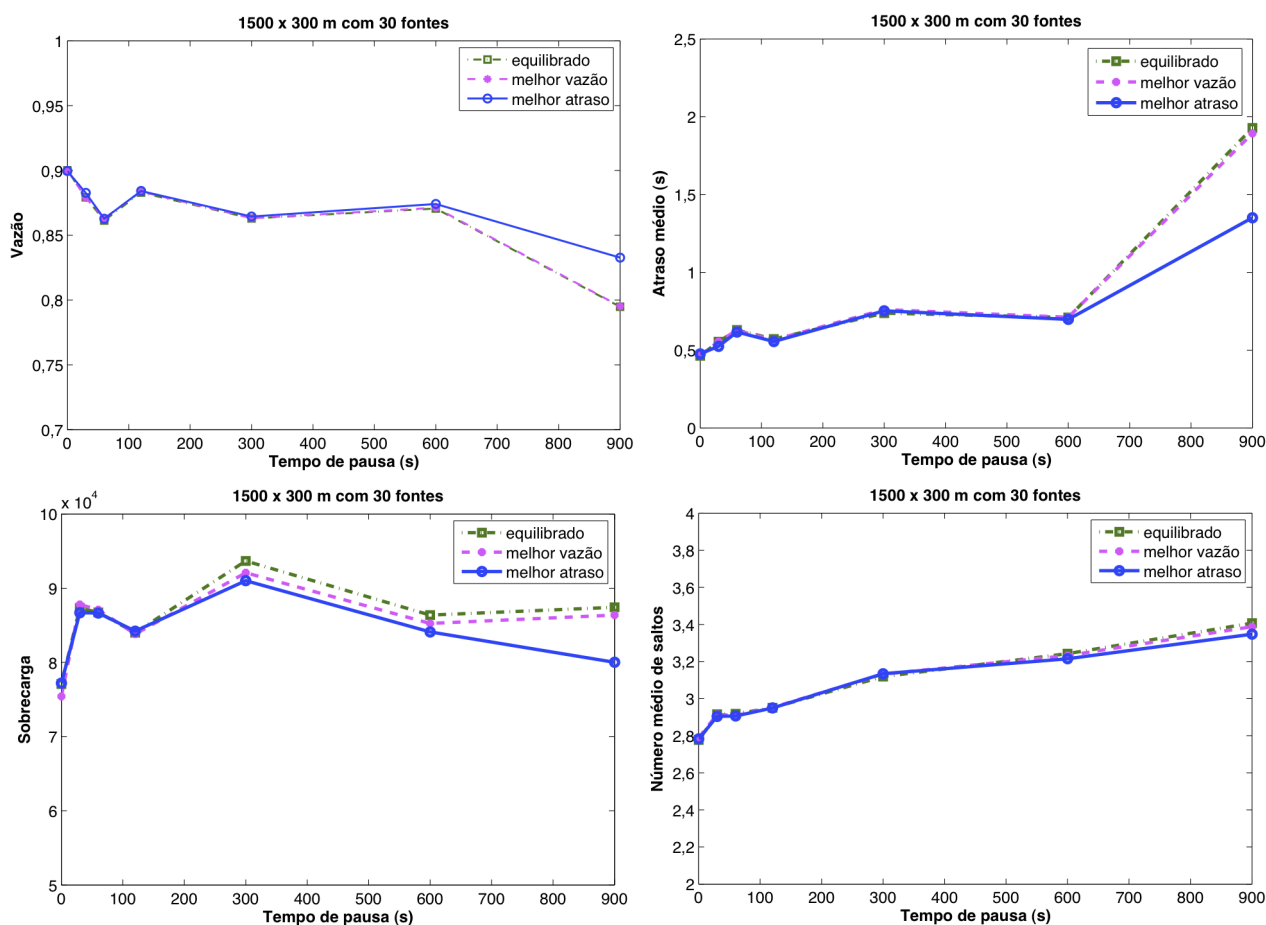


Figura 13: Gráficos comparativos dos resultados das simulações para cenários com 30 conexões considerando as melhores soluções do MOPSO-CDR para atraso e vazão e a mais equilibrada.

5.3 Comparação do AODV original com a versão PSO e a versão MOPSO-CDR

Nesta seção, o algoritmo padrão do AODV é chamado de AODV original, o algoritmo com as modificações sugeridas em [Pereira 2010], de AODV-PSO e o algoritmo com a proposta deste trabalho, de AODV-MOPSO.

Como, em [Pereira 2010], os cenários simulados só consideravam 30 conexões fim-a-fim, os cenários com 10 conexões simulados para este trabalho são comparados apenas como o AODV original. Para esse caso, as Figuras 14 e 15 mostram que os resultados obtidos com o AODV utilizando o MOPSO-CDR são muito próximos dos obtidos com o AODV original, uma vez que o uso do AODV original já consegue valores muito bons para o atraso médio e vazão da rede. Ou seja, perdendo ou ganhando, não há uma diferença substancial ou considerável entre as duas abordagens.

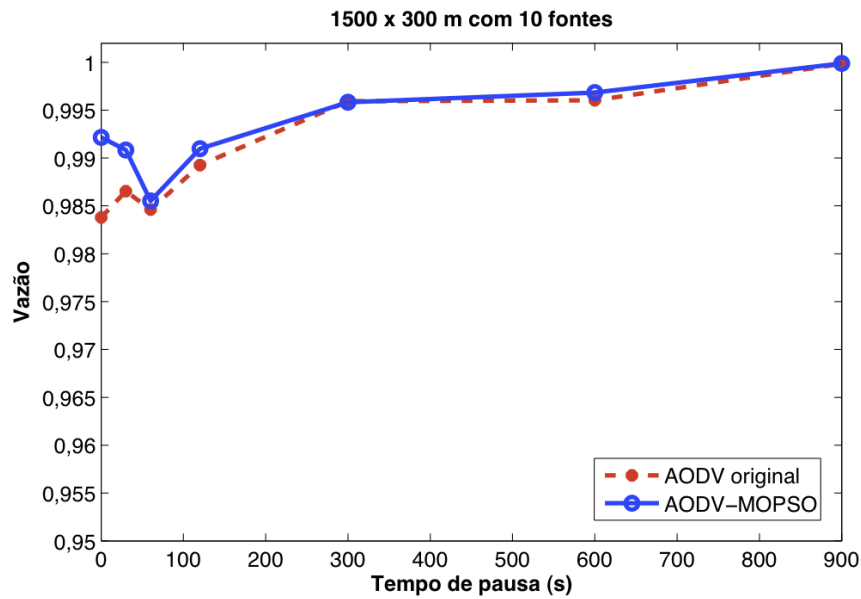


Figura 14: Vazão em função do tempo de pausa do AODV original e do AODV utilizando o MOPSO-CDR para 10 conexões.

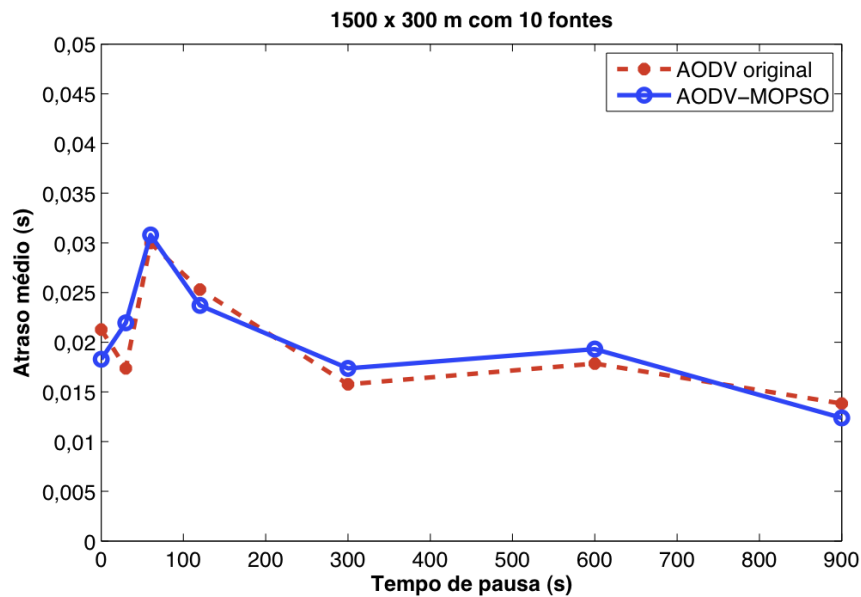


Figura 15: Atraso médio em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.

Entretanto, vale notar que esse não é o caso para os resultados da vazão em cenários com muita mobilidade. Nesse caso, o uso do AODV-MOPSO faz com que a vazão dos pacotes seja superior a 99% quando antes era de aproximadamente 98%. Em termos práticos, isso mostra que se, a cada segundo da simulação, pelo menos um nó transmitir pacotes de dados, dos 3600 pacotes que seriam transmitidos pela rede durante os 900 segundos, apenas cerca

de 30 pacotes não chegariam ao destino usando o AODV-MOPSO contra uma perda de mais de 60 com o AODV original, ou seja, aproximadamente duas vezes mais pacotes perdidos.

Apesar da pequena diferença entre os gráficos do AODV e do AODV-MOPSO, quando o foco são as métricas otimizadas, as medidas secundárias apresentaram um bom ganho com a modificação realizada no AODV. Isso pode ser visto nos gráficos das Figuras 16 e 17. O número médio de saltos diminuiu, em média, 10%; enquanto a sobrecarga da rede, para tempos de pausa que vão de 0 até 120, diminuiu significativamente com ganhos que vão desde 18%, para tempo de pausa 120, até cerca de 33%, para tempo de pausa 0.

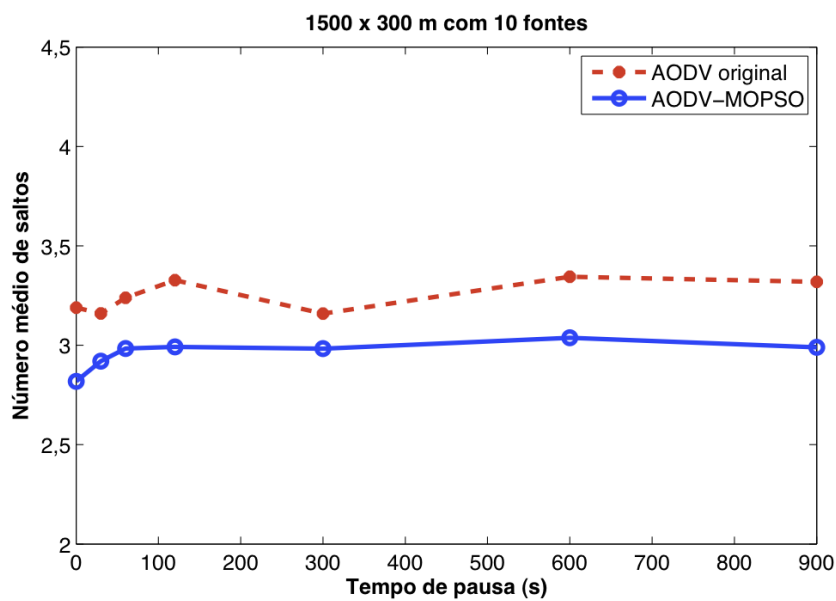


Figura 16: Número médio de saltos em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.

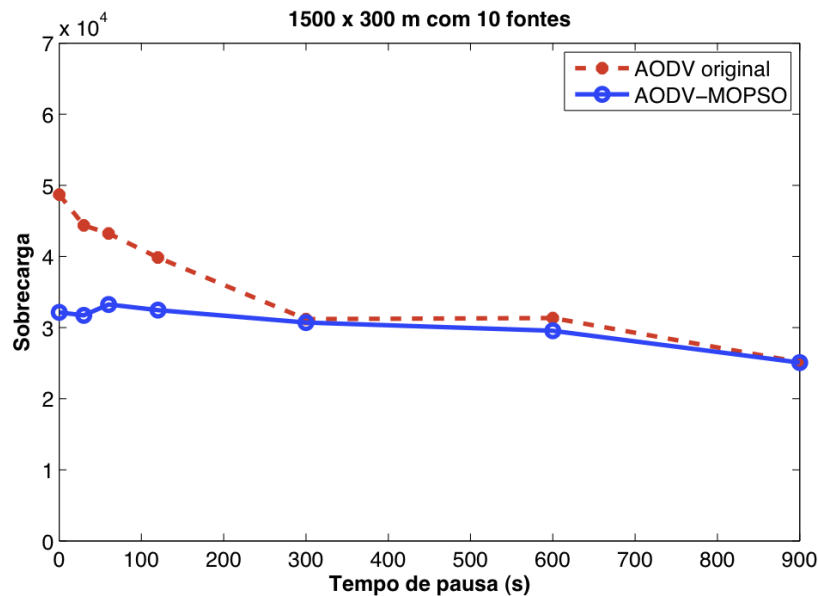


Figura 17: Sobrecarga em função do tempo de pausa do AODV original e do AODV-MOPSO para 10 conexões.

Para os cenários com 30 conexões fim-a-fim, o AODV-MOPSO é comparado tanto com o AODV original quanto com o AODV-PSO.

O AODV-MOPSO mostrou ser melhor do que as outras duas abordagens especialmente para tempos de pausa que vão de 0 a 120, quando se trata das métricas otimizadas, apresentadas nas Figuras 18 e 19. Para esses cenários com grande mobilidade, o atraso obtido com a nova abordagem, assim como com o AODV-PSO, melhorou em relação ao AODV original, mas obteve melhoras que chegaram a até 50%, superando o AODV-PSO nesses casos. Em se tratando de vazão, as melhorias promovidas pelo AODV-MOPSO para tempos de pausa até 120 superaram nitidamente as promovidas pelo AODV-PSO, com a vazão para um cenário de mobilidade total dos nós chegando a ter 90% dos seus pacotes entregues ao destino.

Apesar de ter mostrado melhores resultados do que o AODV original e o AODV-PSO, para tempos de pausa até 120, em alguns casos, o AODV-MOPSO perdeu para o AODV-PSO e, para o tempo de pausa 900, ele perdeu tanto para o AODV-PSO quanto para o AODV original. Isso deve-se ao fato da diversidade de soluções do *Pareto Front* ser mínima para o cenário onde não há nenhuma mobilidade da rede.

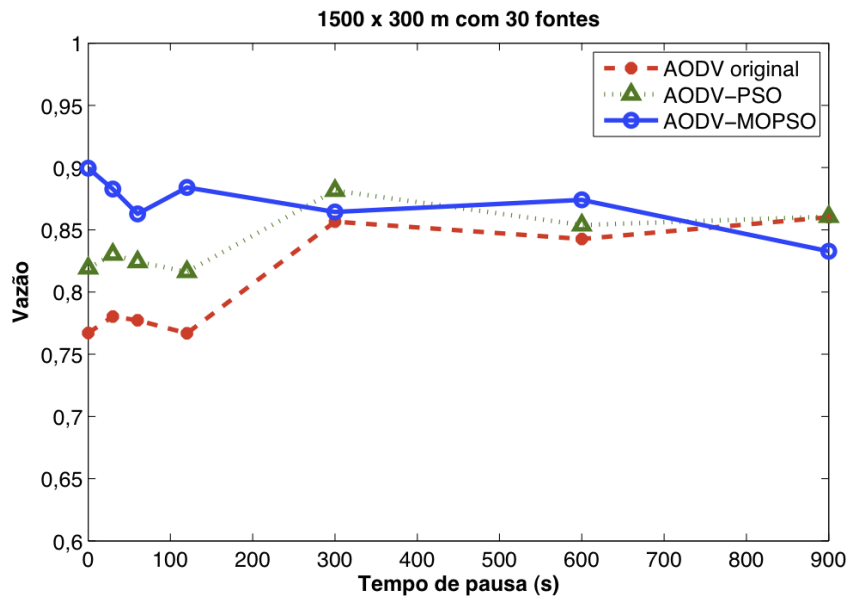


Figura 18: Vazão em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.

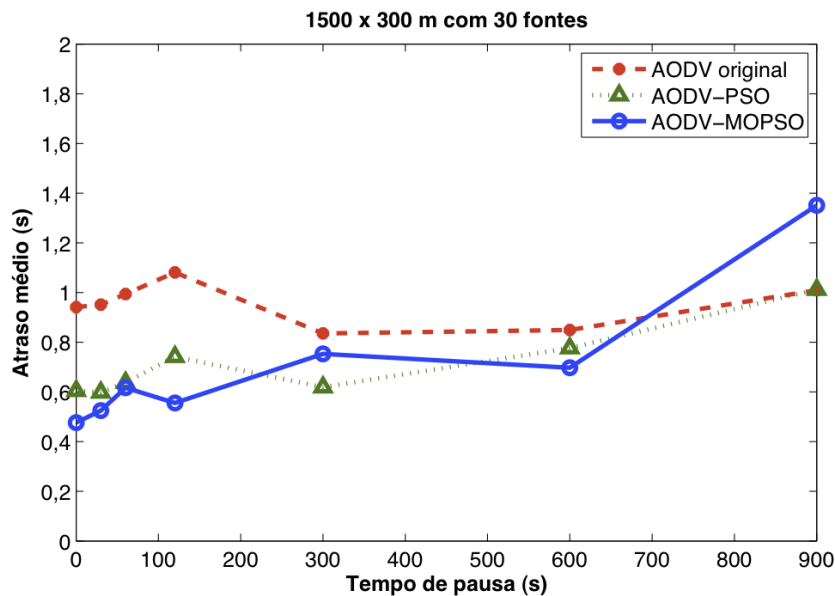


Figura 19: Atraso médio em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.

No que diz respeito ao número médio de saltos, os resultados do AODV-MOPSO foram semelhantes aos do AODV original e do AODV-PSO, como pode ser visto na Figura 20. Já se tratando da sobrecarga, a Figura 21, de forma similar ao que aconteceu com a vazão e o atraso médio, os resultados foram melhores do que as outras duas abordagens para tempos de pausa até 120. Entretanto, nesse caso, os percentuais das melhorias obtidas vão desde 33% e 23% até 43% e 34%, em comparação com o AODV original e o AODV-PSO, respectivamente.

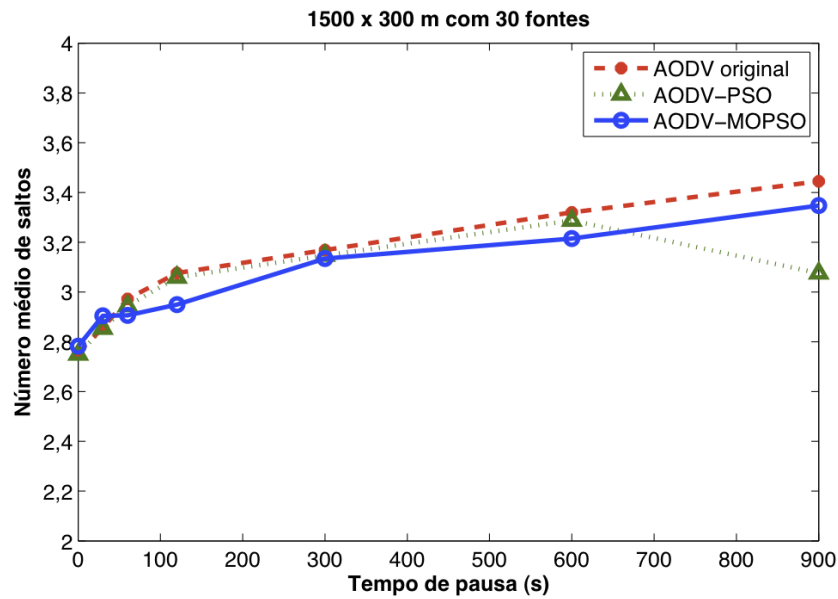


Figura 20: Número médio de saltos em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.

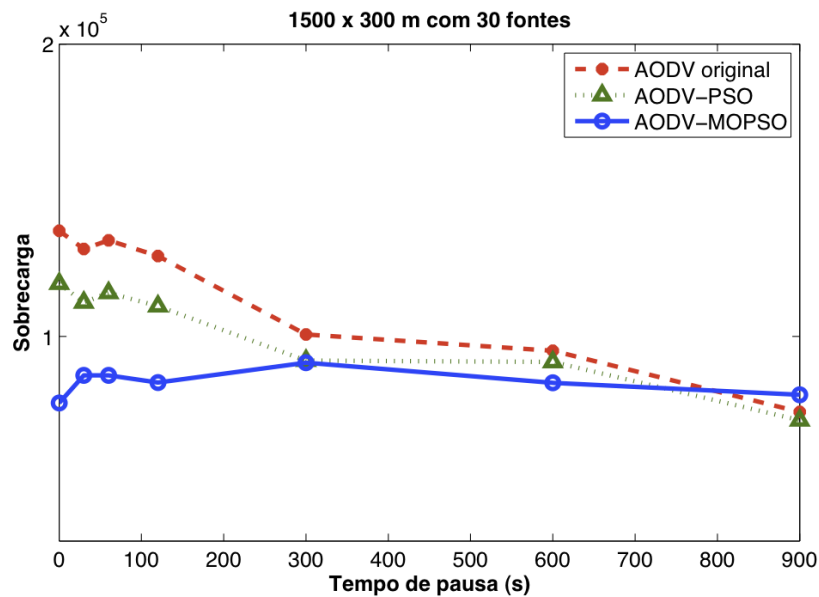


Figura 21: Sobrecarga em função do tempo de pausa do AODV original, do AODV-PSO e do AODV-MOPSO para 30 conexões.

Todos os resultados aqui apresentados comprovam que a abordagem multiobjetiva na otimização do AODV foi uma evolução muito válida, além de natural, da otimização de único objetivo proposta em [Pereira 2010].

6 Conclusões

Este trabalho apresentou uma nova proposta de otimização do protocolo de roteamento de redes *ad hoc* AODV. Uma vez que a aplicação da técnica de Inteligência Computacional, PSO, ao AODV, proposta por [Pereira 2010], obteve resultados interessantes na otimização de redes *ad hoc* que utilizam o AODV como seu protocolo de roteamento, o foco deste trabalho foi estender essa proposta realizando uma otimização multiobjetiva por enxame de partículas, através do MOPSO-CDR, para otimizar simultaneamente a vazão e o atraso médio.

A proposta considerou cenários de rede com características que são comumente usadas no estado-da-arte em pesquisas sobre o roteamento nas redes *ad hoc*. Diante dos cenários considerados, pode-se dividi-los em dois grupos: os com 10 conexões fim-a-fim e os com 30 conexões fim-a-fim.

Para os cenários com 10 conexões, não foi possível analisar os resultados obtidos comparando-os com os obtidos pela aplicação da PSO ao AODV. Comparando-os com os resultados do algoritmo original do AODV, pode-se concluir que, uma vez que os resultados do algoritmo original já são muito bons para a vazão e para o atraso médio da rede, não houve uma diferença significativa em aplicar ou não o MOPSO-CDR. A exceção fica para a vazão em cenários com mobilidade dos nós total ou quase total: aplicando o MOPSO-CDR ao AODV diminui pela metade a perda de pacotes na rede.

Entretanto, para as métricas secundárias, as melhorias apresentadas foram maiores. O número de saltos, em média diminuiu 10% e a sobrecarga, em cenários com alta mobilidade dos nós, chegou a ser reduzida em até 1/3.

Por sua vez, para os cenários com 30 conexões, foi possível comparar os resultados aqui obtidos tanto com o algoritmo original do AODV quanto com a modificação que faz uso da PSO. Nesse caso, o uso do MOPSO-CDR obteve excelentes resultados para cenários com muita mobilidade quando as métricas analisadas eram atraso médio e vazão, o que já era esperado, pois a proposta de [Pereira 2010] também conseguiu seus melhores resultados nessas condições. Entretanto, o atraso médio, quando comparado ao AODV original, foi reduzido

até pela metade, o que representa um resultado superior também ao uso da PSO.

No que diz respeito à vazão da rede, a proposta desse trabalho conseguiu obter uma vazão de até 90% dos pacotes transmitidos, enquanto o uso da PSO conseguiu uma média de cerca de 80% e o AODV padrão de 75%.

As métricas secundárias obtiveram resultados distintos. Enquanto os resultados do número de saltos foi equivalente aos das outras abordagens, a sobrecarga da rede comportou-se de maneira similar à vazão e ao atraso médio, obtendo maiores melhorias para redes com muita mobilidade, e diminuiu mais de 40% em comparação com o AODV original e mais de 1/3 quando comparada ao que foi obtido usando a PSO.

Dessa forma, pode-se concluir que, de maneira geral, a aplicação de técnicas de Inteligência Computacional para a otimização de redes *ad hoc* que utilizam o AODV como protocolo de roteamento, é especialmente eficiente em cenários mais complexos ou caóticos, onde o número de conexões é maior, resultando em mais tráfego na rede, e a mobilidade dos nós é alta, o que implica em mais quebras de enlace ao longo de rotas ativas.

Além disso, as conclusões obtidas a partir da análise dos resultados mostraram-se uma excelente contribuição para promover um uso ainda maior de técnicas de Computação Inteligente no campo das Redes de Telecomunicações.

6.1 **Trabalhos Futuros**

Como trabalhos futuros, a proposta desenvolvida durante o trabalho de conclusão de curso pode ser expandida para que o MOPSO-CDR otimize simultaneamente, além da vazão e do atraso, o número médio de saltos que um pacote percorre e a sobrecarga na rede. Também é interessante simular cenários com áreas maiores, como 1500 x 1500 metros, por exemplo, e para mais cenários além dos 10 aqui considerados para cada tempo de pausa e para cada quantidade de conexões.

Além disso, outros tipos específicos de redes *ad hoc*, como as redes de sensores, onde um alto nível de vazão e um baixo atraso médio não são tão fundamentais quanto o consumo eficiente de energia, podem ser simuladas levando em consideração suas próprias métricas na hora da otimização.

Referências

- [Abramsson 1970]ABRAMSSON, N. The ALOHA system - another alternative for computer communications. *AFIPS Conference Proceedings*, FJCC, v. 37, p. 695–702, 1970.
- [Atzori, Iera e Morabito 2010]ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. *Computer Networks*, Elsevier, v. 54, p. 2787–2805, 2010.
- [BETTSTETTER e WAGNER]BETTSTETTER, C.; WAGNER, C. The Spatial Node Distribution of the Random Waypoint Mobility Model. p. 4158.
- [Broch et al. 1998]BROCH, J. et al. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ACM, p. 85–97, 1998.
- [Chakeres e Belding-Royer 2002]CHAKERES, I. D.; BELDING-ROYER, E. M. The Utility of Hello Messages for Determining Link Connectivity. *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, IEEE, v. 2, p. 504–508, 2002.
- [Coello, Pulido e Lechuga 2004]COELLO, C.; PULIDO, G.; LECHUGA, M. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 8, n. 3, p. 256–279, 2004.
- [Das, Perkins e Royer 2000]DAS, S. R.; PERKINS, C. E.; ROYER, E. M. Performance Comparison of Two On-demand Routing protocols for Ad Hoc Networks. *Proceedings of INFOCOM 2000*, IEEE, p. 3–12, 2000.
- [FALL 2009]FALL, K. V. K. The Network Simulator - NS-2. 2009.
- [Filho 2008]FILHO, R. J. U. de M. S. *Análise do Escalonamento de Redes Ad Hoc IEEE 802.11 através de medidas de Vazão e Atraso usando o NS-2*. Monografia (Trabalho de Conclusão de Curso) — Universidade de Pernambuco, 2008.
- [Frodigh, Johansson e Larsson 2000]FRODIGH, M.; JOHANSSON, P.; LARSSON, P. Wireless ad hoc networking - The art of networking without a network. *Ericsson Review*, Ericsson, n. 4, p. 248–263, 2000.
- [Gupta et al. 2010]GUPTA, P. et al. Optimized Use of Battery Power in Wireless Ad hoc Networks. *Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT)*, IEEE, 2010.
- [Haas e Pearlman 2000]HAAS, J. Z.; PEARLMAN, M. R. Status of this Memo: The Zone Routing Protocol (ZRP) for Ad Hoc Networks. 2000.
- [Jacquet et al. 2001]JACQUET, P. et al. Optimized Link State Routing Protocol for Ad Hoc Networks. *Proceedings of IEEE INMIC*, IEEE, p. 62–68, 2001.

- [J. Jubin e J.D. Tornow 1987] J. JUBIN; J. D. TORNOW. The DARPA packet radio network protocol. *Proceedings Of the IEEE*, IEEE, v. 75, n. 1, p. 21–32, 1987.
- [Joa-Ng e Lu 1999] JOA-NG, M.; LU, I.-T. A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, IEEE, v. 17, n. 8, p. 1415–1425, 1999.
- [Johansson et al. 1999] JOHANSSON, P. et al. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad hoc Networks. *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking*, ACM, 1999.
- [Johnson e Maltz 1996] JOHNSON, D. B.; MALTZ, D. A. Dynamic Source Routing in Ad Hoc Wireless Networks. In: IMIELINSKI, T.; KORTH, H. (Ed.). *Mobile Computing*. [S.l.]: Kluwer Academic Publishers, 1996. p. 153–181.
- [Kai, Neng e Ai-fang 2005] KAI, Z.; NENG, W.; AI-FANG, L. A new AODV based clustering routing protocol. *Proceedings of Wireless Communications, Networking and Mobile Computing (WCNM)*, IEEE, 2005.
- [Kennedy e Eberhart 1995] KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, IEEE, v. 4, p. 1942–1948, 1995.
- [Liu e Lin 2005] LIU, J.; LIN, C.-H. R. RBR: refinement-based route maintenance protocol in wireless ad hoc networks. *Computer Communications*, Elsevier, v. 28, n. 8, p. 908–920, 2005.
- [Ma e Kim 2009] MA, Y.; KIM, D. I. Centralized and Distributed Optimization of Ad-Hoc Cognitive Radio Network. *Proceedings of the the IEEE "GLOBECOM"2009*, IEEE, 2009.
- [Murthy e Manoj 2004] MURTHY, C. S. R.; MANOJ, B. *Ad Hoc Wireless Networks: Architectures and Protocols*. 1. ed. [S.l.]: Prentice Hall, 2004.
- [Nedjah, Coelho e Mourelle 2010] NEDJAH, N.; COELHO, L. dos S.; MOURELLE, L. de Macedo de (Ed.). *Multi-Objective Swarm Intelligent Systems. Theory Experiences*, v. 261. [S.l.]: Springer, 2010.
- [Noishiki, Yokota e Idoue 2005] NOISHIKI, Y.; YOKOTA, H.; IDOUE, A. Efficient on-demand route establishment methods for dense ad-hoc networks. *Proceedings of Autonomous and De centralized Systems (ISADS)*, IEEE, 2005.
- [Pan, Chuang e Wang 2005] PAN, M.; CHUANG, S.-Y.; WANG, S.-D. Local Repair Mechanisms for On-Demand Routing in Mobile Ad hoc Networks. *Proceedings of the 11th Pacific Rim International Symposium on Dependable Computing*, 2005.
- [Particle Swarm Optimization: Source Codes & Demos] Particle Swarm Optimization: Source Codes & Demos. Disponível em <www.swarmintelligence.org/codes.php>. Acesso em: 15 de novembro 2010.
- [Pereira 2010] PEREIRA, N. C. V. N. *Uma Proposta para Melhorar o Protocolo de Roteamento AODV em Redes Ad Hoc Utilizando Otimização por Enxame de Partículas*. Dissertação (Mestrado em Engenharia da Computação) — Universidade de Pernambuco, 2010.

- [Pereira, Buarque e Moraes 2009]PEREIRA, N. C. V. N.; BUARQUE, C. E.; MORAES, R. M. de. Um Estudo do Desempenho dos Mecanismos de Recuperação de Falha de Rotas do Protocolo AODV em Redes Ad Hoc sem Fio. *Proceedings of the 8th International Information and Telecommunication Technologies Symposium (I2TS)*, p. 1093–1097, 2009.
- [Perkins, Belding-Royer e Das 2003]PERKINS, C.; BELDING-ROYER, E.; DAS, S. Ad hoc On-Demand Distance Vector (AODV) Routing: RFC 3561. Internet Engineering Task Force, Network Working Group, 2003.
- [Perkins e Bhagwat 1994]PERKINS, C. E.; BHAGWAT, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Proceedings of the ACM SIGCOMM '94*, ACM, p. 234–244, 1994.
- [Perkins e Royer 1997]PERKINS, C. E.; ROYER, E. M. Ad-hoc On-Demand Distance Vector Routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, IEEE, p. 90–100, 1997.
- [Raquel e Naval 2005]RAQUEL, C. R.; NAVAL, J. P. C. An effective use of crowding distance in multiobjective particle swarm optimization. *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, ACM, p. 257–264, 2005.
- [Reyes-Sierra e Coello 2006]REYES-SIERRA, M.; COELLO, C. A. C. Multi-Objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, v. 2, n. 3, p. 287–308, 2006.
- [Santana 2009]SANTANA, R. A. *A Multiple Objective Particle Swarm Optimization Approach Using Crowding Distance And Roulette Wheel*. Dissertação (Mestrado em Engenharia da Computação) — Universidade de Pernambuco, 2009.
- [The Network Simulator - ns-2]The Network Simulator - ns-2. Disponível em <www.isi.edu/nsnam/ns/>. Acesso em: 10 de novembro 2010.
- [Tsou et al. 2007]TSOU, C. et al. Using Crowding Distance to Improve Multi-Objective PSO with Local Search. In: CHAN, F. T. S.; TIWARI, M. K. (Ed.). *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*. [S.l.]: Itech Education and Publishing, 2007. p. 77–86.
- [Wireless Simulation Scenarios in ns-2]Wireless Simulation Scenarios in ns-2. Disponível em <http://www.winlab.rutgers.edu/~zhibinwu/html/ns2_wireless_scene.htm>. Acesso em: 15 de novembro 2010.
- [Youn et al. 2006]YOUN, J. et al. Quick local repair scheme using adaptive promiscuous mode in mobile ad hoc networks. *Journal of Networks*, Academy Publisher, v. 1, n. 1, p. 1–11, 2006.
- [Yousefi, Mlousavi e Fathy 2006]YOUSSEFI, S.; MIOUSAVI, M. S.; FATHY, M. Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. *2006 6th International Conference on ITS Telecommunications Proceedings*, IEEE, p. 761–766, 2006.
- [Zitzler 1999]ZITZLER, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Tese (PhD Dissertation) — ETH Zurich, 1999.