

Detecção de Agrupamentos de Portadores na Monitoração Telemática de Tornozeleiras Eletrônicas

Paulo Lima Machado; Rafael Timóteo de Sousa Júnior; Robson de Oliveira Albuquerque; Daniel Alves da Silva; Valério Aymoré Martins; Flávio Elias Gomes de Deus

Resumo — Sistemas de monitoramento por tornozeleiras eletrônicas comumente oferecem detecção de violação de zonas de circulação permitida aos portadores. Entretanto, tais sistemas não identificam o agrupamento de indivíduos monitorados, evento que pode representar um risco real de cometimento de novas infrações. Este trabalho propõe algoritmos para identificação desses agrupamentos e reunião de informações adicionais para auxiliar análises de riscos nesse contexto.

Palavras-Chave — *Sistemas de telecomunicações de aplicação da lei (LETS), Geoprocessamento, Monitoramento, Algoritmos.*

Abstract — Monitoring systems for electronic ankle bracelets commonly detect violations of circulation areas permitted to holders. However, such systems do not identify the grouping of individuals monitored, an event that could represent a real risk of further offenses. This paper proposes algorithms to identify such groups and gather information to support risk analysis in this context.

Keywords — *Law Enforcement Telecommunications Systems (LETS), Geoprocessing, Monitoring, Algorithms.*

I. INTRODUÇÃO

A utilização de tornozeleiras eletrônicas em pessoas investigadas e condenadas tem sido aplicada pelas autoridades brasileiras para tentar reduzir o encarceramento em massa no país. Segundo dados de dezembro de 2013 do Departamento Penitenciário Nacional (DEPEN) do Ministério da Justiça, o Brasil possui uma das maiores populações prisionais do mundo, com 581.507 detentos [1].

A utilização de meios alternativos de vigilância dos sentenciados tem-se mostrado uma viável alternativa para fiscalização da execução de penas, levando ao desenvolvimento de inovações na forma de controle de indivíduos que infringem normas penais. Uma dessas alternativas é o monitoramento por satélite de detentos portadores de tornozeleiras eletrônicas, forma de vigilância que se mostrou efetiva nos Estados Unidos e na Europa [2].

A Fig. 1 ilustra uma das arquiteturas utilizada em sistemas de monitoramento de sentenciados com tornozeleiras

Paulo Lima Machado, Rafael Timóteo de Sousa Júnior, Robson de Oliveira Albuquerque, Daniel Alves da Silva e Valério Aymoré Martins, Flávio Elias Gomes de Deus, Departamento de Engenharia Elétrica, Universidade de Brasília - UnB, Brasília-DF, Brasil, paulo.machado@aluno.unb.br, desousa@unb.br, robson@redes.unb.br, daniel.alves@redes.unb.br, valeriomartins@unb.br, flavioelias@unb.br. Os autores agradecem às Agências brasileiras de pesquisa, desenvolvimento e inovação CAPES (Projeto FORTE 23038.007604/2014-69) e FAPDF (Protocolo 11454.54.34054.17052016), e à Defensoria Pública da União (Acordo de Cooperação 30/2014 DPU-FUB), pelo suporte ao presente trabalho.

eletrônicas, sistemas estes em que as coordenadas coletadas pelos dispositivos (1) são enviadas pela rede de transmissão de dados GSM/GPRS através de operadoras de telefonia celular (2) com destino à Central de Monitoramento (3). Esta última tem a função de processar os dados geográficos provenientes das tornozeleiras e gerar relatórios e alertas para que as autoridades competentes tomem providências de acordo com as políticas de cada sistema. A informação gerada mais comum é a indicação de violação de zonas proibidas ao monitorado.

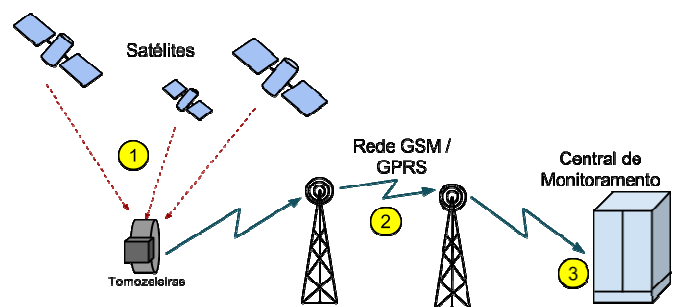


Fig. 1. Sistema de monitoramento por tornozeleiras.

O presente artigo considera a possibilidade de a Central de Monitoramento também fornecer serviços agregados referentes a dados sobre formação de agrupamentos de indivíduos monitorados, com base na detecção de proximidade das coordenadas fornecidas pelas tornozeleiras. Além de mostrar os grupos e localização, é também possível consolidar informações sobre intervalo de tempo em que cada grupo permanece reunido, quantidade de elementos, frequência e horário dos encontros. Tais informações têm o potencial de contribuir em análises de riscos envolvendo ações preventivas por parte de órgãos de segurança pública.

Sistemas de telecomunicações de aplicação da lei (*Law Enforcement Telecommunications Systems* – LETS) devem levar em conta o risco real representado por grupos específicos, levando em consideração fatores como o nível de periculosidade de seus elementos e os tipos de infrações cometidas por cada um, entre outros. Assim, é relevante desenhar algoritmos capazes de fornecer dados para subsidiar a análise de risco e a tomada de decisão nesse contexto. Objetiva-se desse modo auxiliar a geração de alertas informando prováveis formações de motins, preparação para ações criminosas, entre outras atividades suspeitas. Identificar antecipadamente tais riscos mostra-se particularmente importante em épocas de grandes eventos sociais como, por exemplo, as Olimpíadas deste ano no Brasil.

A contribuição focal do presente artigo é a concepção de um conjunto de algoritmos articulados, constituindo um

modelo sistêmico capaz de processar os dados provenientes da rede para: 1) verificar proximidades (Detecção de Pares), 2) agrupar dispositivos próximos em *clusters* (Detecção de Grupos) e 3) registrar duração de grupos e quantidade média de elementos agrupados (Detecção de Riscos). Contribuições adicionais são descritas quanto aos aspectos de implementação e de desempenho desses algoritmos. Tais algoritmos são aplicáveis em outras situações como, por exemplo, agrupamentos de animais monitorados em florestas.

O restante deste artigo tem a seguinte organização: A Seção II discute os trabalhos relacionados. A Seção III caracteriza o problema da detecção de agrupamentos e introduz o modelo sistêmico da solução proposta. A Seção IV apresenta a análise do modelo sistêmico e dos resultados algoritmos em ambiente simulado. Já a Seção V conclui este artigo e apresenta possíveis trabalhos futuros.

II. TRABALHOS RELACIONADOS

Artigos que tratam do processamento geográfico de pontos e a identificação de *clusters* baseiam-se em geral na procura de concentração de pontos analisando sua distribuição. Porém, não levam em consideração a necessidade específica de identificar indivíduos reunidos a partir de pontos que estão dentro de uma distância mínima que caracterize a reunião. Sem tal consideração, a concentração detectada pode referir-se a pontos separados por distâncias até da ordem de quilômetros e não de alguns poucos metros, o que foge ao conceito de reunião ou encontro de pessoas monitoradas.

Liu *et al.* [3] aborda algoritmos que identificam *clusters* de objetos classificados em categorias, considerando aspectos puramente geográficos ou outros atributos associados. Aborda principalmente o algoritmo “*density-based spatial clustering*” (DBSC) que identifica *clusters* usando tanto a proximidade espacial quanto a similaridade de atributos. O DBSC envolve a construção de relacionamentos de proximidades entre pontos obtidos por meio da triangulação de *Delaunay* [4].

Carlino [5] argumenta sobre a influência da proximidade física de laboratórios de pesquisa e desenvolvimento (P&D) na repercussão do conhecimento na área de concentração dos mesmos. Para tanto, compara a localização dos laboratórios no território dos EUA com o registro de patentes na mesma área, evidenciando sua ligação. Então, aborda uma maneira de medir a extensão da concentração espacial das atividades dos laboratórios e definir o *cluster* formado pelos laboratórios próximos, considerando um círculo em volta de cada ponto de localização, com um raio inicial de um quarto de milha. Em seguida, enumera a quantidade de pontos dentro do círculo. Como resultado, muitos círculos se sobrepõem formando o *cluster* a ser analisado e comparado com o registro de patentes.

Já o algoritmo DBSCAN [6] busca a identificação de *clusters* com diversos tipos de objetos geográficos (pontos, polígonos, linhas, etc.). Cada grupo adjacente em um dado raio contém pelo menos um número mínimo de pontos, ou seja, sua densidade extrapola algum limite, o que deixa claro que o processamento é executado ponto a ponto utilizando a relação de distância entre pontos conforme também proposto no presente trabalho. O artigo propõe estimar o valor da distância para diferenciar a ideia de concentração de pontos da ideia de pontos dispersos, fora da concentração (ruídos).

Os trabalhos supracitados, até por estarem desvinculados da questão dos LETS, não atendem a requisitos do problema abordado no presente artigo, ou seja: 1) não consideram a

evolução do grupo no tempo, identificando a duração da concentração de pontos e o tamanho (quantidade de pontos) do grupo, 2) não possuem tempo limite de processamento e identificação dos *clusters*, e 3) alguns algoritmos não impõem um limite de distância mínima entre pontos dos *clusters*.

Por outro lado, o trabalho em [7] trata das questões de criação e manutenção de uma base de dados em PostgreSQL com a extensão PostGIS, armazenando dados geográficos transmitidos por sensores localizados na Itália. Descreve as etapas de criação da base de dados e as tabelas necessárias para demonstração e armazenamento de dados geográficos. O presente artigo complementa tal análise com maior detalhamento do banco de dados e requisitos de implementação para validar os algoritmos aqui apresentados.

III. DESCRIÇÃO DO PROBLEMA E MODELO SISTÊMICO

Dadas as coordenadas GPS coletadas de pontos de presença de dispositivos monitorados em uma rede como a da Fig. 1, a caracterização de uma reunião é o problema para o qual este artigo propõe uma possível solução, tal como descrito abaixo.

A. Problema

A concentração de tais pontos em uma área geográfica não necessariamente indica um agrupamento de indivíduos reunidos, na ótica do monitoramento de pessoas para aplicação da lei. Por exemplo, para certa concentração de pontos em uma área do espaço geográfico, cuja menor distância encontrada é de 1 km, não se pode afirmar que os monitorados estão realmente reunidos apesar de observar-se a concentração inclusive visualmente no mapa. De fato, para que dois ou mais monitorados sejam considerados reunidos em um grupo, deve haver uma distância mínima estabelecida entre os pontos que os representam.

Os algoritmos propostos neste trabalho devem realizar o processamento de pontos geográficos para identificação de agrupamento de monitorados considerando tal distância, além de atualizar uma base com dados complementares como o tempo de duração da formação do grupo e a respectiva quantidade de elementos.

Todas as etapas deverão ser executadas em um período de tempo tal que não ultrapasse uma janela de processamento estabelecida em função do requisito de aplicação da lei. Tal janela é parametrizada e fixada inicialmente em um minuto. Além disso, trata-se de um limite de desempenho, pois, caso esse tempo seja ultrapassado, há o risco de acúmulo com o processamento da janela anterior ou sobrecarregando o equipamento responsável pelo processamento, ou levando à perda de informação.

B. Modelo Sistêmico

A distância mínima característica de uma reunião, limiar este utilizado no processamento, deve levar em consideração a margem de erro (ϵ) inerente aos equipamentos de geolocalização com GPS. Segundo [8], a precisão dos dispositivos GPS é pouco menor que 10 metros. Assim, tendo em consideração tal margem de erro, aplica-se a Eq. 1 para definir o limiar de reunião (lr). Ou seja, na prática, para efeito de cálculo, dois pontos quaisquer separados por distância inferior a 20 metros serão considerados monitorados próximos.

$$lr = 2 \cdot \lceil \epsilon_{min} \rceil \quad (1)$$

A Fig. 2 apresenta o modelo sistêmico proposto neste artigo que divide o processamento em 3 etapas: 1) recebe da rede os pontos enviados pelos dispositivos e calcula os pontos próximos (Detecção de Pares); 2) agrupa os pontos próximos em *clusters* (Detecção de Grupos); e 3) acrescenta a duração do grupo e a quantidade de participantes (Detecção de Riscos).

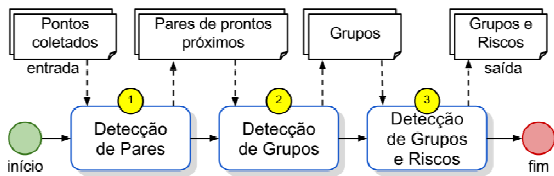


Fig. 2. Etapas do Processamento.

Algoritmo para Detecção de Pares

Nesta etapa, correspondente ao processo 1 da Fig. 2, os pontos são comparados entre si e os pares cuja distância seja menor ou igual à *lr* metros são identificados como pares de pontos próximos e adicionados a uma lista que comporá a saída do algoritmo.

À medida que aumenta a quantidade de pontos pertencente ao conjunto coletado, a quantidade de comparações necessárias para identificação de pares próximos também aumenta. A quantidade de comparações é dada pela fórmula da análise combinatória por combinação simples (Eq. 2), onde *p* é a quantidade de elementos da combinação sempre 2 a 2 (pares) e *n* é a quantidade de pontos da amostra.

$$C_{n,p} = \frac{n!}{p!(n-p)!} = \frac{n \cdot (n-1) \cdot (n-2)!}{2 \cdot (n-2)!} = \frac{n \cdot (n-1)}{2} \quad (2)$$

Esta etapa é, em termos computacionais, umas das mais onerosas, pois implica na comparação entre todos os pontos da amostra para identificar os pontos próximos.

O Algoritmo de Detecção de Pares (Fig. 3) tem como entrada uma lista de pontos coletados com a estrutura detalhada na Tabela I e seus passos encontram-se na Tabela II.

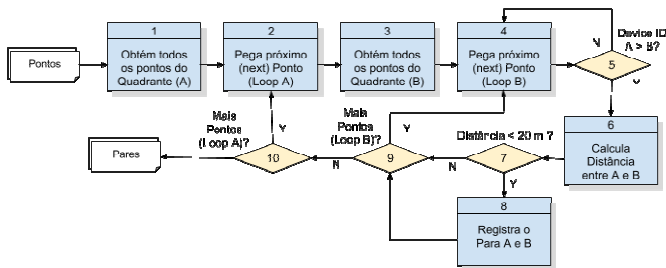


Fig. 3. Algoritmo 1 - Detecção de Pares.

Algoritmo para Detecção de Grupos

Nesta etapa, correspondente ao processo 2 da Fig. 2, os pares de pontos próximos identificados e armazenados no item anterior são lidos e agrupados. A ideia básica para agrupamento consiste em localizar situações onde o ponto A está próximo de B e o ponto B está próximo de C, e então A, B e C formam um grupo de monitorados reunidos. O Algoritmo de Detecção de Grupos é apresentado na Fig. 4 e seus passos encontram-se na Tabela III.

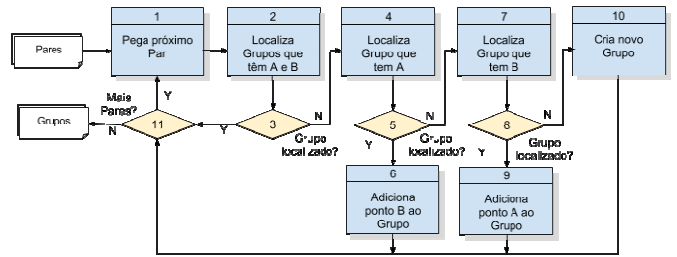


Fig. 4. Algoritmo 2 - Detecção de Grupos.

TABELA I. ENTRADA DO ALGORITMO DE DETECÇÃO DE PARES

Atributo	Tipo	Descrição
Device ID	Inteiro	Número identificador do dispositivo
Timestamp	Data Hora	Data, hora e minuto da coleta do ponto
Point	Coordenada Geográfica	Ponto geográfico composto pela Latitude e Longitude

TABELA II. PASSOS DO ALGORITMO DE DETECÇÃO DE PARES

#	Descrição
1	Como é necessário comparar os pontos entre si, o processo inicia um <i>loop</i> considerando todos os pontos coletados.
2	Pega cada ponto obtido do item anterior referente ao <i>Loop A</i> .
3	Varre novamente todos os pontos coletados (<i>Loop B</i>) a serem comparados com os pontos do <i>Loop A</i> .
4	Pega cada ponto obtido do item anterior referente ao <i>Loop B</i> .
5	Este filtro evita que o cálculo da distância de A em relação a B seja repetido para B em relação a A. Se A for maior que B, ignora este par e volta para o Passo 4.
6	Utiliza a função de geoprocessamento, obtendo a distância em metros de A em relação a B.
7	Se a distância for maior que <i>lr</i> metros, então os pontos A e B não estão próximos e o fluxo segue para o próximo ponto do <i>Loop B</i> .
8	Se os pontos A e B estão a <i>lr</i> metros ou menos de distância, então são considerados próximos e são armazenados para a etapa de agrupamento.
9	Se houver mais pontos relativos ao <i>Loop B</i> para serem comparados, então desvia o fluxo para pegar o próximo ponto B.
10	Se houver mais pontos relativos ao <i>Loop A</i> para serem comparados, então desvia o fluxo para pegar o próximo A. Se não houver, o processamento de comparação estará concluído gerando como resultado registros com pares de dispositivos próximos.

TABELA III. PASSOS DO ALGORITMO DE DETECÇÃO DE GRUPOS

#	Descrição
1	Obtém cada par do conjunto obtido da etapa anterior.
2	Verifica se há algum grupo das iterações anteriores que já tenha os pontos A e B para eliminar alguma eventual repetição de grupo.
3	Se for identificado um grupo com os 2 pontos, então nada precisa ser feito e o loop deve prosseguir par próximo par.
4	Verifica se há algum grupo que tenha pelo menos o ponto A.
5	Se o grupo foi localizado, então não possui o ponto B.
6	Adiciona o ponto B ao grupo localizado no item anterior.
7	Verifica se há algum grupo que tenha pelo menos o ponto B.
8	Se o grupo foi localizado, então não possui o ponto A.
9	Adiciona o ponto A ao grupo localizado no item anterior.
10	Se não foi localizado grupo que contenha A ou B, então um grupo

#	Descrição
	novo deve ser criado com o par A e B. Este grupo poderá então ser acrescido de novos pontos conforme as iterações posteriores.
11	Se ainda há par a ser processado, então pegar o próximo par. Se não há, encerra o algoritmo.

A saída (output) desta etapa é uma lista de grupos com o momento em que foram coletados os pontos (*timestamp*) e uma lista de dispositivos dos grupos.

Algoritmo para Detecção de Riscos

Para inserir ou atualizar dados adicionais referentes à duração dos grupos (processo 3 da Fig. 2) é necessário registrar o momento (*timestamp* inicial) em que o grupo foi identificado pela primeira vez, continuar processando as amostras seguintes (próximos *timestamps*), até o momento da dispersão dos indivíduos monitorados (*timestamp* final) conforme Fig. 5.a.

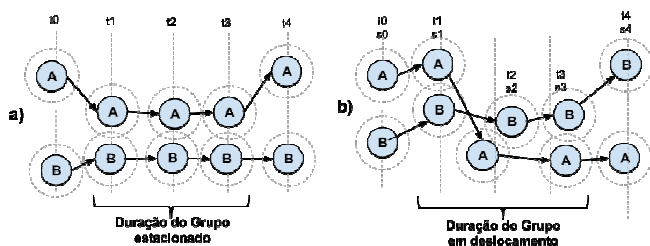


Fig. 5. Mensuração da Duração do Grupo.

As Fig. 5.a e 5.b ilustram a mensuração do tempo de existência dos grupos estando eles estacionados em ou em deslocamento. O Algoritmo de Detecção de Riscos é apresentado na Fig. 6 e seus passos, na Tabela IV.

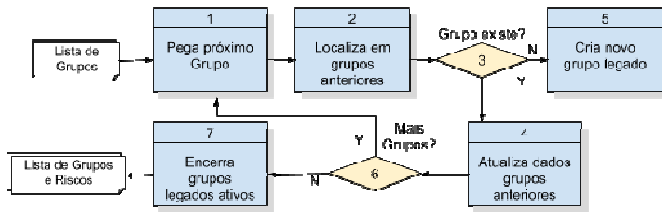


Fig. 6. Algoritmo 3 - Detecção de Riscos

TABELA IV. PASSOS DO ALGORITMO DE DETECÇÃO DE RISCOS

#	Descrição
1	Obtém cada grupo obtido no processamento atual.
2	No conjunto de grupos anteriores ativos, identificam-se aqueles que possuem pelo menos 50% dos elementos em comum com o grupo identificado. O parâmetro percentual 50% é uma escolha arbitrária a ser estudada em trabalhos futuros, não havendo espaço para um estudo no presente trabalho.
3	Se não encontrou nenhum grupo compatível, então desvia o fluxo para criação de um novo grupo.
4	Atualiza a quantidade de processamentos de pontos pelo qual já passou o grupo e a soma de quantidade de elementos em cada um dos processamentos. Estes dois atributos fornecem os dados necessários para cálculo da média de elementos no grupo durante sua existência.
5	Cria um novo grupo considerando o atributo <i>Processing_Count</i> como 1, o atributo <i>Start_Timestamp</i> como sendo o Timestamp do processamento atual e o atributo <i>Elements_Count</i> a mesma quantidade de elementos do grupo.
6	Se houver mais grupos a serem processados, então desvia o fluxo para pegar o próximo grupo. Se não houver, então encerra grupos vigentes, conforme próximo passo 7.

#	Descrição
7	Grupos vigentes que não foram identificados no processamento atual devem ser desativados. Isto é feito atualizando a data <i>End_Timestamp</i> com o valor correspondente ao processamento anterior, última vez que o grupo foi identificado. Como resultado do processamento, será gerado um conjunto de grupos e sua respectiva duração e quantidade de elementos, conforme a estrutura da Tabela 5, para utilização no próximo processamento.

Como resultado (output) desta etapa, uma estrutura de dados deve ser armazenada e/ou atualizada com os elementos especificados na Tabela V. Tal estrutura é então repassada para a etapa de análise de risco e também armazenada de forma que seja atualizada pelos futuros processamentos de pontos.

TABELA V. GRUPOS E RISCOS - OUTPUT

Atributo	Tipo	Descrição
<i>Group ID</i>	Inteiro	Identifica unicamente o Grupo
<i>Start Timestamp</i>	Data Hora	Data/Hora que o Grupo se reuniu.
<i>End Timestamp</i>	Data Hora	Data/Hora que o Grupo se dispersou
<i>Processing Count</i>	Inteiro	Quantidade processamentos que o Grupo foi detectado
<i>Devices Count</i>	Inteiro	Soma das quantidades de elementos do Grupo para cálculo da média
<i>Devices ID List</i>	Lista de Coordenadas Geográficas	Lista dos dispositivos que estiveram dentro do Grupo.

IV. RESULTADOS E ANÁLISES

Para validar a aplicação dos algoritmos apresentados neste trabalho foi usada uma base de dados simulada com aproximadamente 10.000 dispositivos em um SGBD PostgreSQL versão 9.4.

O equipamento utilizado para simular o processamento dos algoritmos possui a configuração detalhada na Tabela VI.

TABELA VI. EQUIPAMENTO UTILIZADO NOS TESTES

Recurso	Especificação
Processador	Core i5-2467M 1.6 GHz (<i>dual core with hyperthread</i>)
Memória	8 Gigabytes
Disco Rígido	Hard Disk 516 GBytes 5400 RPM
Sist. Operacional	Ubuntu 14.04.4
SGBD	PostgreSQL 9.4

A simulação de grupos foi realizada através da criação de variantes de alguns trajetos obtidos por equipamentos GPS. Novos trajetos foram projetados através do deslocamento horizontal e vertical dos pontos originais no espaço geográfico e pela inversão das latitudes e longitudes simulando novos pontos. Como resultado, foram gerados três conjuntos de 10.000 pontos cada. Tais conjuntos correspondem a três amostras em intervalos de um minuto, simulando a coleta de pontos das tornezeiras no dia 24/05/2015, equivalente respectivamente aos horários: i) 12:00; ii) 12:01; e iii) 12:02.

Nessa massa de dados, as coordenadas dos pontos estão distribuídas em uma área total de 146 km². Da forma como foi simulada, a dispersão dos pontos no espaço geográfico difere de uma amostra real, por apresentar maior homogeneidade e,

por isso, gerando maior ocorrência de pontos próximos. Ainda assim, é possível a aplicação dos algoritmos para avaliação do desempenho, uma vez que a estrutura dos dados não se altera. Além disso, considera-se que essa massa de dados foi filtrada, eliminando ocorrência de registros duplicados, ausentes ou fora de sequência. O tratamento dos filtros, por sua complexidade, não é tratado neste trabalho.

Para fins de simulação, foi inicialmente criado o cenário A e posteriormente o cenário B, devido ao problema de desempenho encontrado no cenário A. Todas as execuções foram individualmente repetidas 10 vezes registrando-se neste artigo o respectivo tempo médio de execução. Em todos os casos, foi controlada a execução correta dos algoritmos, pela verificação dos resultados obtidos na detecção de pares, grupos e riscos, cujos valores encontram-se à Tabela VII.

TABELA VII. EQUIPAMENTO UTILIZADO NOS TESTES

Resultados da Execução dos Algoritmos	Amostras		
	12:00	12:01	12:02
Pares Detectados	9.103	9.580	14.686
Grupos Detectados	1.673	1.762	1.854
Riscos Detectados (grupos ativos identificados)	406		

A. Cenário com os algoritmos em PG/plSQL

Aproveitando que a base de simulação dos pontos encontra-se em um SGBD PostgreSQL [9], os três algoritmos especificados foram desenvolvidos na Linguagem PG/plSQL [10], com a utilização de funções da extensão PostGIS para o cálculo de distância entre os pontos. Os algoritmos foram desenvolvidos em funções distintas e executados separadamente, obtendo o tempo médio apresentado na Fig. 7.

Constata-se na Fig. 7 que o tempo de resposta do processamento do primeiro algoritmo ficou bem acima dos demais, em função da necessidade de comparação entre os pontos, o que requer uma grande quantidade de cálculos de distância entre os pontos. Tal tempo ficou acima do limite imposto de 1 minuto, o que motivou a construção do segundo cenário utilizando a linguagem C.

B. Cenário com os algoritmos em Linguagem C

Neste cenário, os algoritmos 2 e 3 permaneceram em Linguagem PG/plSQL devido aos seus baixos tempos de repostas, enquanto o algoritmo 1 foi reescrito em linguagem C.

Os tempos de processamento da mesma quantidade de pontos com a rotina do Algoritmo 1 em PG/plSQL e em Linguagem C são apresentados na Fig. 8.

V. CONCLUSÃO E TRABALHOS FUTUROS

Os algoritmos propostos se mostraram efetivos na captura do fenômeno de formação de grupos de pessoas portadoras de tornozeleiras eletrônicas dotadas de GPS.

A necessidade de cuidados de implementação é determinante da obtenção de um desempenho apropriado, em particular à monitoração voltada à aplicação da lei. Uma implementação integral em linguagem PG/plSQL, resultou em tempo total somando os três algoritmos acima do limite apropriado, inviabilizando a aplicação prática daquele cenário.

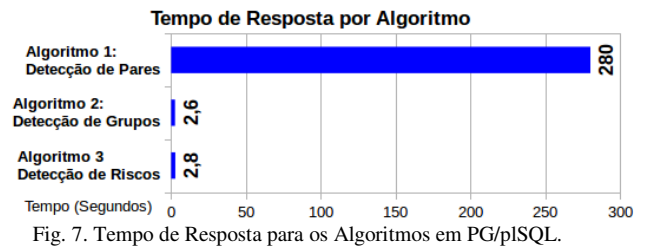


Fig. 7. Tempo de Resposta para os Algoritmos em PG/plSQL.

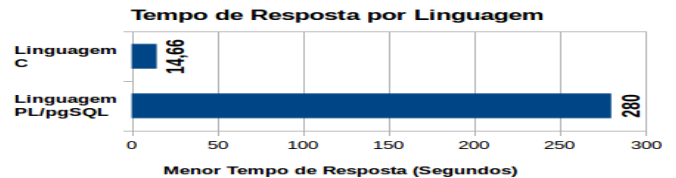


Fig. 8. Tempos de Resposta do Algoritmo 1 em PG/plSQL e em C.

Já o desenvolvimento do Algoritmo 1 em Linguagem C levou a um tempo de resposta compatível com o limiar de janela de tempo de processamento de 1 minuto.

Há necessidade de um estudo mais profundo da parametrização desse limiar, assim como de outros fixados arbitrariamente neste trabalho, haja vista um eventual requisito de redução do tempo entre as amostras. Se eventualmente houver redução do intervalo de amostras para 30 segundos, metade do original, o tempo de processamento em Linguagem C consumirá aproximadamente 30% da janela disponível.

Também como trabalho futuro, para melhorar o tempo de resposta na execução do Algoritmo 1, sugere-se alterá-lo de forma que o mesmo possa ter seu processamento paralelizado aproveitando os múltiplos núcleos de CPUs disponíveis nos computadores modernos, inclusive no equipamento utilizado nos testes descritos neste trabalho.

A comparação dos resultados aqui apresentados com aqueles de técnicas semelhantes abordadas por outros autores é tema a ser explorado na continuidade deste trabalho.

REFERÊNCIAS

- [1] DEPEN – Departamento Penitenciário Nacional. URL <http://dados.mj.gov.br/dataset/infopen-levantamento-nacional-de-informacoes-penitenciarias>.
- [2] R. M. Barbosa, *O Monitoramento Eletrônico para Presos de Baixa Periculosidade*, Universidade Católica de Brasília, 2010, Pág.11.
- [3] Q. Liu, M. Deng, Y. Shi, J. Wang. *A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity*. Computers & Geosciences, 2012.
- [4] L. P. Chew, *Algorithmica*, Springer, 1989, Volume 4, Pág. 97.
- [5] G. A. Carlino, J. K Carr. *Clusters of knowledge: R&D proximity and the spillover effect*. Business Review - Federal Reserve Bank of Philadelphia, 2013.
- [6] S. Louhichi, M. Gzara, H. B. Abdallah. *A density based algorithm for discovering clusters with varied density*. In Computer Applications and Information Systems (WCCAIS), 2014 World Congress on. IEEE, 2014.
- [7] F. Urbano, H. Dettki. *Storing tracking data in an advanced database platform (postgresql)*. In Spatial Database for GPS Wildlife Tracking Data, Pág. 9–24. Springer, 2014.
- [8] NSTB/WAAS T&E Team, *Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report*, Federal Aviation Administration – FAA, 2014.
- [9] G. Smith, *PostgreSQL 9.0 High Performance*, Packt Publishing, 2010.
- [10] R. O. Obe, L. S. Hsu. *PostGis In Action*, Manning, 2011.