Gabriel Leonardo Pedote

Avaliação do Impacto da Seleção de Partições Base em Ensemble Multiobjetivo

Sorocaba, SP

23 de Fevereiro de 2018

Gabriel Leonardo Pedote

Avaliação do Impacto da Seleção de Partições Base em Ensemble Multiobjetivo

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Computação Científica e Inteligência Computacional.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Orientadora: Profa. Dra. Tiemi Christine Sakata

Sorocaba, SP

23 de Fevereiro de 2018

Pedote, Gabriel Leonardo

Avaliação do Impacto da Seleção de Partições Base em Ensemble Multiobjetivo / Gabriel Leonardo Pedote. -- 2018.
72 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador: Profa. Dra. Tiemi Christine Sakata
Banca examinadora: Profa. Dra. Flávia Cristina Bernardini, Prof. Dr. Murilo Coelho Naldi
Bibliografia

1. Seleção de Partições. 2. Ensemble Multiobjetivo. 3. Diversidade e Qualidade. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).
DADOS FORNECIDOS PELO(A) AUTOR(A)
Bibliotecário(a) Responsável: Maria Aparecida de Lourdes Mariano – CRB/8 6979

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Gabriel Leonardo Pedote, realizada em 23/02/2018:

Profa. Dra. Tiemi Christine Sakata
UFSCar

Profa. Dra. Flávia Cristina Bernardini
UFF

Prof. Dr. Murilo Coelho Naldi
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Flávia Cristina Bernardini e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Profa. Dra. Tiemi Christine Sakata

Dedico o presente trabalho aos meus pais, grande parte de quem sou é fruto da dedicação e amor de vocês. Sou eternamente grato e sempre estarei em débito.

Agradecimentos

Agradeço,

À minha orientadora, Profa. Dra. Tiemi Christine Sakata, pelos conselhos, paciência e comprometimento comigo.

À minha família que sempre me apoiou durante toda minha vida e em minhas decisões; os momentos que passamos juntos estarão sempre comigo.

Aos professores da UFSCar, em especial a Profa. Dra. Katti Faceli e Prof. Dr. Tiago Almeida pelos conselhos e contribuições valiosas a esse trabalho.

Aos membros da banca do presente trabalho, Profa. Dra. Flávia Bernardini e Prof. Dr. Murilo Naldi pelas sugestões, contribuições e comentários.

Aos colegas da UFSCar, em especial a Vanessa Antunes e o Randal Gasparini, por todo o suporte e companheirismo.

Ao Programa Pós-graduação em Ciência da Computação da UFSCar Sorocaba, por essa grande oportunidade de crescimento profissional e pessoal.

À toda a equipe da UFSCar, pela diligência, atenção e respeito com que sempre fui tratado.

À CAPES pelo apoio financeiro.

Have patience with everything unresolved in your heart and to try to love the questions themselves as if they were locked rooms or books written in a very foreign language. Don't search for the answers, which could not be given to you now, because you would not be able to live them. And the point is, to live everything. Live the questions now. Perhaps then, someday far in the future, you will gradually, without even noticing it, live your way into the answer.

(Letters to a Young Poet, Rainer Maria Rilke)

Resumo

Agrupamento não-supervisionado não é um processo trivial, uma vez que não existe conhecimento prévio e dados reais são tipicamente complexos e multifacetados. Para piorar, algoritmos tradicionais, em grande parte, descrevem os dados em questão sob uma única perspectiva. E ao fazê-lo, impõem sérias limitações no que pode ser extraído com a análise. Para mais, alterações na parametrização e tratativas de pré-processamento podem alterar radicalmente o resultado final, seja evidenciando ou escondendo uma possível pluralidade de significados presente nos dados. Para amenizar alguns desses problemas, técnicas que consideram múltiplas partições surgiram; como, por exemplo, *ensemble clustering*. Contudo, para utilizá-las deve-se ter em mente a qualidade e a diversidade dessas múltiplas partições, já que ambas características se mostraram fortemente relacionadas a performance dessas técnicas. Uma das formas de favorecer a qualidade e a diversidade dessas múltiplas partições — e tender a melhores resultados — é selecionar e considerar apenas um subconjunto das partições disponíveis, descartando as que não possuem esses dois atributos. Para tal, diversos métodos de seleção de partições foram propostos e aplicados com sucesso na literatura. Nesta dissertação, expandimos essa discussão avaliando o impacto de diferentes métodos de seleção pertencentes ao estado da arte em um contexto inédito, o de *ensemble* multiobjetivo. Nesse novo contexto, as análises realizadas indicam ganhos em duas importantes frentes — ambas destacadas na literatura do *framework* aqui tratado: **(i)** os resultados são mais concisos, o que, na prática, facilita a interpretação manual dos mesmos por especialistas, e **(ii)** são obtidos com uma fração do esforço computacional, isto é, com maior rapidez e onerando menos recursos. Tudo isso sem implicar perda de qualidade nos resultados finais.

Palavras-chaves: Seleção de Partições. Ensemble Multiobjetivo. Diversidade e Qualidade.

Abstract

Unsupervised data clustering is not a trivial process, as no previous knowledge is available and real data is often complex and multi-faceted. To make matters worse, traditionally, clustering aims to describe the data being explored under a single perspective. However, it is broadly known that in several cases this approach imposes serious limitations on what could be extracted with the analysis. Furthermore, changes in parameters and preprocessing techniques can dramatically change the final result, either by evidencing or by hiding a possible plural meaning presented in the data. To tackle some of these issues, recent efforts that build knowledge considering multiple partitions as base, such as ensemble clustering, emerged. However, special care must be taken in the composition of those partitions, as their quality and diversity proved to be closely related to their performances. To enhance the quality and diversity of those multiple partitions — and provide better results —, a number of methods to evaluate and select a subset of the partitions have been proposed and successfully applied. In this work, we expand this discussion by evaluating the impact of some of the state-of-the-art selection methods in the novel context of multi-objective cluster ensemble. In this novel context, our analysis show improvements in two important issues: **(i)** the results are more concise, which facilitates posterior manual analysis, and **(ii)** are obtained with less computational effort. All of that without affecting the quality of the results.

Key-words: Partition Selection. Multi-objective Clustering Ensemble. Diversity and Quality.

Lista de ilustrações

Lista de tabelas

Índice de algoritmos

Lista de abreviaturas e siglas

AL	*Average-Link*
AMI	*Adjusted Mutual Information*
ANOVA	*Analysis of Variance*
ARI	*Adjusted Rand Index*
ASA	*Automatic Selection Algorithm*
ASS	*Alternative Simplified Silhouette*
BRP	*Best Rank Position*
CAS	*Cluster And Select*
CE	*Centroid-Link*
CO	*Complete-Link*
CRI	*Combination of Relative Indexes*
DB	*Davies-Bouldin*
FILTA	Filtered Meta-clustering
HSS	*Hybrid Selection Strategy*
KM	*K-means*
MCHPF	*Multi-objective Clustering with Hierarchical Partitions Fusions*
MCLA	*Meta-Clustering Algorithm*
MOCK	*Multi-Objective Clustering with automatic K-determination*
MOCLE	*Multi-objective Clustering Ensemble*
NSGA-II	*Non-dominated Sorting Genetic Algorithm II*
SIS	*Single Index Selection*
SL	*Single-Link*
SNN	*Shared Nearest Neighbors*

SRD	*Sum of Ranks with Diversity*
SR	*Sum of Ranks*
SS	*Simplified Silhouette*
NMI	*Normalized Mutual Information*
VRC	*Davies-Bouldin*

Lista de símbolos

μ_i	Centroide do *cluster* c_i
Π	Conjunto de partições
Π_C	Conjunto inicial de uma seleção
Π_E	Conjunto de estruturas conhecidas em um *dataset*
Π_I	Conjunto de partições base do MOCLE
Π_R	Conjunto reduzido de uma seleção
Π_S	Conjunto de saída do MOCLE
Π_{Rep}	Repositório de Partições
π^{Ei}	i-ésima estrutura conhecida
π^i	i-ésima partição
$\mid . \mid$	Cardinalidade de um conjunto
ARI_m	*Adjusted Rand Index* (ARI) médio
c_i	i-ésimo *cluster* de uma partição
$M_{p \times p}$	Matriz de tamanho p
Q, H	funções para avaliar a qualidade de uma partição
d	Número de atributos em um conjunto de dados
$d(x_i, x_j)$	Distância entre os objetos x_i e x_j
K	Número de *clusters* de uma partição
K^{Ei}	Número de *clusters* da i-ésima estrutura conhecida
K^{max}	Número máximo de *clusters*
K^{min}	Número mínimo de *clusters*
n	Número de objetos (dimensões) em um conjunto de dados
n_c	Número de *clusters* de uma partição

NN	Número de vizinhos mais próximos a um dado objeto
nn_{ij}	é o $j-$ésimo vizinho mais próximo ao objeto x_i
s	Número de partições a serem selecionadas
x_i	i-ésimo objeto do conjunto de dados X
X	Conjunto de dados

Sumário

1 Introdução

Técnicas de agrupamento são importantes ferramentas para análise exploratória de dados. Elas auxiliam no processo de obtenção do conhecimento através da classificação de objetos em *clusters* (grupos). Essa classificação se dá tradicionalmente de forma não supervisionada, isto é, sem a necessidade de conhecimento prévio sobre possíveis estruturas presentes nos dados (JAIN; DUBES, 1988)[1]. Tais características possibilitam a aplicação dessas técnicas no auxílio a resolução de uma ampla gama de problemas em diversas áreas, como: biologia, medicina, visão computacional, *marketing* e detecção de anomalias em redes de computadores (JAIN; DUBES, 1988; JAIN, 2010).

Apesar dos diversos casos de sucesso em múltiplas áreas, aplicar agrupamento de dados não é uma tarefa trivial. Isso porque, em sua maioria, algoritmos tradicionais buscam descrever os dados em questão sob uma única perspectiva (FACELI et al., 2011). E ao fazê-lo, impõem sérias limitações no que pode ser extraído com a análise. Especialmente em casos reais, onde os dados são frequentemente complexos e podem ser naturalmente agrupados de várias formas (BAILEY, 2013). Além disso, alterações nos parâmetros desses algoritmos (e.g., número de *clusters*) e a aplicação de técnicas de pré-processamento (e.g., seleção de atributos) podem alterar radicalmente o resultado final, seja evidenciando ou escondendo uma possível pluralidade de significados presente nos dados (JAIN; DUBES, 1988; MULLER et al., 2010).

A Figura 1 exemplifica como dados, ainda que simples, podem ser multifacetados. Nela consta um exemplo de um *dataset* (i.e., conjunto de dados) para segmentação de mercado — que pode ser utilizada em *marketing*, por exemplo. Nesse exemplo, é possível obter três agrupamentos (i.e., modelos) que descrevem esses dados, ao considerar diferentes combinações de atributos. Esse exemplo demonstra como, sem cuidados especiais, algoritmos tradicionais de agrupamento são limitados em buscar apenas um agrupamento. Para mais, vale ressaltar, assim como Muller et al. (2010) fez, que há uma série de aplicações reais em que os dados podem ser multifacetados sem necessariamente ser através de subconjuntos de atributos. Exemplos disso são aplicações em categorização de texto, em que documentos são agrupados por similaridade de tópicos tratados. Nesse tipo de análise, cada documento pode pertencer a mais de um grupo, em diferentes níveis de refinamento. Por exemplo, um documento sobre aprendizado de máquina pode ser, em um nível de refinamento maior, também agrupado por subtópicos, como: agrupamento de dados, regressão, etc. Logo, como diferentes refinamentos são necessários, não existe um

[1] Existem técnicas de agrupamento que envolvem conhecimento prévio, como por exemplo, agrupamento semi-supervisionado, porém, nesse trabalho, tratamos apenas de agrupamento como uma técnica não supervisionada.

único agrupamento capaz de mapear esse comportamento.

Figura 1 – Exemplo de *dataset* no qual mais de agrupamento pode ser obtido. Sendo o primeiro agrupamento possível, aquele que considera os atributos idade e renda. E o segundo possível, aquele que considera os atributos pressão arterial e atividade esportiva praticada.

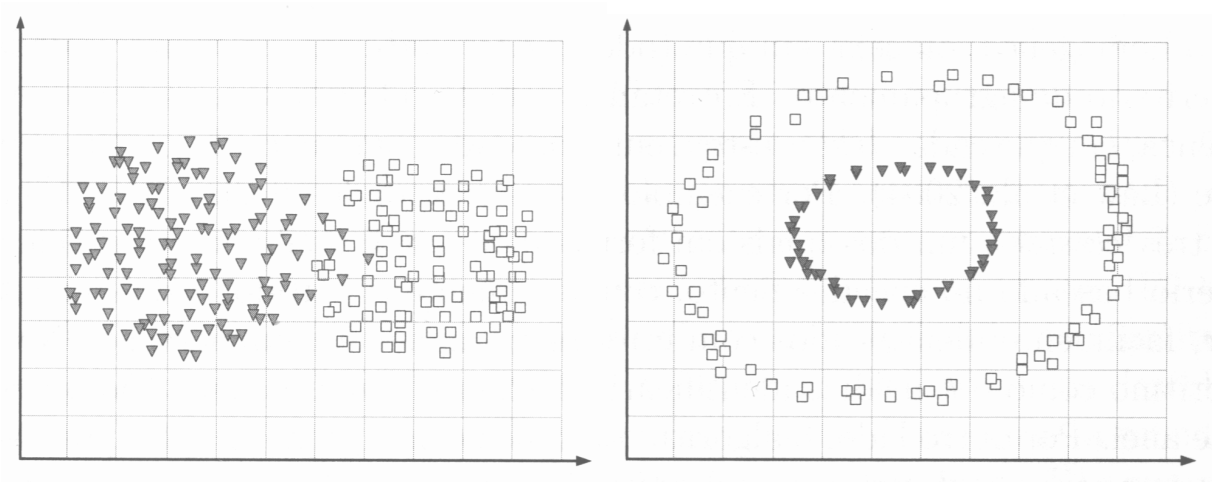
Fonte: adaptado de Muller et al. (2010).

Outro entrave para uma análise de sucesso é o critério de agrupamento de um algoritmo, pois quando esse não está em consonância com as estruturas[2] presentes os dados ou os objetivos do usuário, agrupamentos não significativos tendem a ser obtidos (FACELI et al., 2011). Por exemplo, algoritmos como o *K-means*, que buscam otimizar o critério de compactação, são efetivos para evidenciar *clusters* esféricos e/ou bem separados (e.g., Figura 2a), porém tendem a falhar para estruturas mais complexas (FACELI et al., 2011). Outro exemplo, são algoritmos que identificam encadeamento, como *Single-Link*, que são apropriados para detectar *clusters* de formas arbitrárias (e.g., Figura 2b). Porém, costumam falhar em detectar casos onde há pouca separação espacial entre os *clusters* (e.g., Figura 2a).

Para dificultar ainda mais a questão, uma mesma base de dados pode ser composta por uma estrutura heterogênea de *clusters*, isto é, com *clusters* de diferentes formatos. No entanto, cada algoritmo tradicional, busca uma estrutura homogênea, ou seja, *clusters* em conformidade com os mesmos critérios de agrupamento; logo, não existe um único algoritmo capaz de encontrar os diferentes tipos de agrupamento que possam estar presentes em um *dataset* (FACELI et al., 2011). Na Figura 3a está exemplificado esse cenário. Nela, está ilustrado um *dataset* que é composto de três *clusters*, dois globulares e um anelar. Estruturas assim, desafiam algoritmos como *K-means*, que conseguem identificar os *clusters* globulares, mas falham no anelar (Figura 3b). Por outro lado, também desafiam algoritmos

[2] Uma estrutura, no escopo desse trabalho, se refere ao mesmo conceito de um agrupamento ou partição. A diferença entre eles é que uma estrutura é um agrupamento válido já conhecido dos dados.

Figura 2 – Dados com *clusters* em conformidade com diferentes critérios.



(a) Globular. (b) Anel.

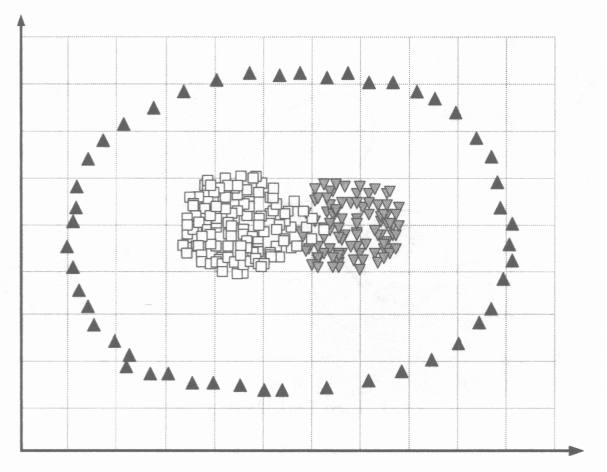
Fonte: (FACELI et al., 2011).

como o *Single-Link* (Figura 3c), que conseguem identificar o anelar, mas falham nos outros dois.

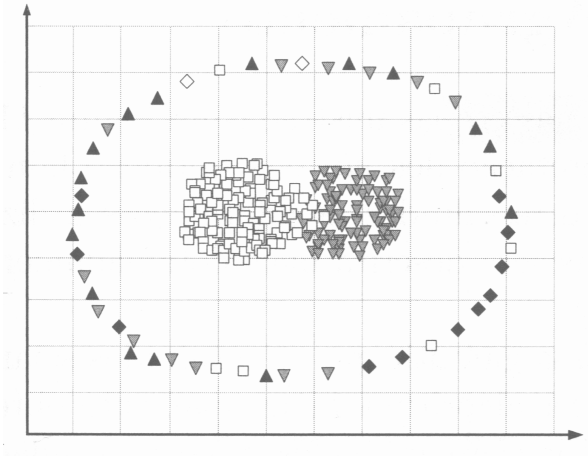
Em suma, há diferentes questões a serem consideradas para um agrupamento significativo. Muitas dessas podem ser tratadas através de etapas manuais de validação, com o uso de heurísticas e conhecimento de um usuário/especialista. Porém, há abordagens mais recentes que lidam melhor com essas questões, automatizando e facilitando a obtenção de agrupamentos significativos. Dentre elas, técnicas que consideram múltiplas visões sobre os dados são as que mais se destacam e mostram-se robustas perante diferentes conformações dos dados (JAIN, 2010; FACELI et al., 2011; ZIMEK; VREEKEN, 2015). Essas múltiplas visões são normalmente obtidas através do uso de um ou mais algoritmos tradicionais de agrupamento, executados repetidamente com diferentes parâmetros e inicializações. Há, no entanto, variadas formas de considerar essas múltiplas perspectivas (ZIMEK; VREEKEN, 2015).

Ensemble clustering, por exemplo, objetiva identificar, através da combinação de múltiplos agrupamentos, apenas uma solução consenso (JAIN, 2010). O intuito é que essa solução seja superior em qualidade e estabilidade aos agrupamentos individuais dos quais ela se origina. Experimentalmente, bons resultados têm sido obtidos, porém *ensembles* também possuem deficiências (HANDL; KNOWLES, 2007; FACELI et al., 2011). Primeiro, porque buscam uma única estrutura e por essa razão, assim como algoritmos tradicionais de agrupamento, são limitados em evidenciar múltiplos significados nos dados. Segundo, porque, como todas as múltiplas visões disponíveis são utilizadas simultaneamente, pode haver influência negativa no resultado final, caso haja presença de um grande número de partições de baixa qualidade (FACELI et al., 2011).

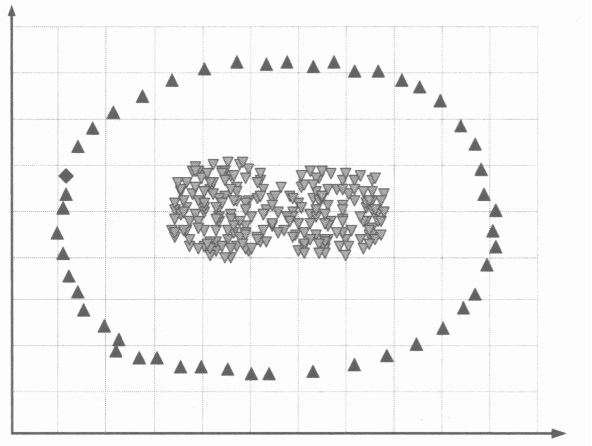
Figura 3 – Conjunto de dados com uma estrutura de *clusters* heterogênea e os resultados dos agrupamentos.



(a) Conjunto de dados heterogêneo.



(b) Agrupamento obtido pelo *K-means*.



(c) Agrupamento obtido pelo *Single-Link*.

Fonte: (FACELI et al., 2011).

Algoritmos evolutivos multiobjetivo são uma alternativa aos *ensembles* (FACELI et al., 2011). Nesse caso, mais de uma solução é pretendida, por meio da otimização de diferentes funções objetivo. Porém, esses algoritmos possuem elevado custo computacional e tendem a resultar em um alto número de partições, incluindo soluções similares (ANTUNES, 2018), o que dificulta a análise manual desses resultados por um usuário/especialista com domínio dos dados.

Já a abordagem *ensemble* multiobjetivo, que combina as duas técnicas anteriores, tenta superar essas dificuldades individuais, aproveitando os benefícios de ambas abordagens (FACELI; CARVALHO; SOUTO, 2008). Não obstante, nessa abordagem, também há oportunidades para melhoras. Amenizar alguns dos problemas dessa abordagem é o foco desse trabalho.

1.1 Objetivos

Para representar a técnica *ensemble* multiobjetivo o *framework Multi-Objective Clustering Ensemble* (MOCLE) foi adotado. Estudos preliminares, em Pedote, Faceli e Sakata (2017), identificaram três oportunidades para melhora. Amenizar esses três problemas, descritos a seguir, é o objetivo do presente trabalho.

Qualidade: Diversos estudos empíricos comparam o MOCLE com diferentes abordagens de *ensemble* e de algoritmos multiobjetivo evolutivos (FACELI, 2006; FACELI; CARVALHO; SOUTO, 2006; FACELI; CARVALHO; SOUTO, 2007a; FACELI; CARVALHO; SOUTO, 2007b; FACELI; CARVALHO; SOUTO, 2008; FACELI et al., 2009; FACELI; SOUTO; CARVALHO, 2010; PIANTONI et al., 2015). Esses estudos demonstram a superioridade do MOCLE em encontrar partições de alta qualidade (do interesse do usuário) em diversos casos. Porém, ainda há espaço para melhorias, já que em alguns casos o MOCLE não inclui, em sua saída, partições de alta qualidade fornecidas como entrada e/ou não as leva em conta em suas combinações (PEDOTE; FACELI; SAKATA, 2017). Isso significa que o MOCLE, em alguns casos, descarta partições que muito provavelmente seriam de interesse ao usuário e não as fornece como parte de seu resultado final. Alterar o *framework* para também incluir essas partições de alta qualidade em sua saída é parte dos objetivos do presente esforço.

Número de partições: O MOCLE consegue produzir conjuntos de resultados mais concisos (i.e., com menor número de agrupamentos) em comparação com algoritmos puramente multiobjetivo evolutivos (FACELI, 2006). O que, na prática, é uma vantagem, pois isso facilita a análise/interpretação manual dos resultados por um usuário/especialista (FACELI, 2006). Porém, embora já conciso, o conjunto de partições geradas, pode ser ainda menor, sem necessariamente implicar perda de qualidade (PEDOTE; FACELI; SAKATA, 2017). Conduzir o *framework* a produzir conjuntos de partições (resultados) mais concisos também integra os objetivos desse trabalho. Vale lembrar ainda, que a facilidade de uso do MOCLE é um dos seus principais objetivos (FACELI, 2006) e que esse objetivo está diretamente alinhado a ele.

Tempo de execução: Um dos maiores impeditivos da aplicação do MOCLE é sua alta complexidade computacional, pois isso resulta em longos períodos de execução e alta demanda de recursos computacionais (FACELI; CARVALHO; SOUTO, 2008). Handl e Knowles (2004) argumentam que em casos complexos de agrupamento, o tempo de execução dessas técnicas são fatores secundários, já que a maior parte do tempo é gasta em processos manuais — tais quais, elaboração, preparação, execução de experimentos e coleta de dados. Porém, o MOCLE, mesmo em *datasets* relativamente pequenos, tende a tomar horas ou dias de processamento, inclusive em ambientes

computacionais robustos[3], o que certamente não torna o tempo um fator secundário nesse caso. Além do tempo, o MOCLE, demanda muitos recursos computacionais para ser executado em um tempo razoável, por isso, tem sua aplicabilidade reduzida em cenários reais, já que nem sempre esses recursos computacionais estão disponíveis. Baseado nessas justificativas, o terceiro objetivo do presente trabalho é reduzir o esforço computacional necessário para execução do *framework* e também o tempo necessário para sua execução. Com isso, o que se pretende é tornar a utilização do MOCLE mais palatável, frente a outras técnicas, em cenários reais.

1.2 Abordagem Proposta

Na tentativa de melhorar os três fatores supracitados, adotamos a mesma abordagem de Pedote, Faceli e Sakata (2017). Isto é, foram utilizados diferentes métodos para pré-processar as múltiplas partições consideradas pelo MOCLE, removendo aquelas que não são de interesse. Diversos trabalhos nos quais o MOCLE se baseia ou que também consideram múltiplos agrupamentos, indicam fortemente que a qualidade e a diversidade dos agrupamentos considerados exercem influência direta na qualidade e robustez dos resultados finais (FERN; LIN, 2008; MULLER et al., 2010; NALDI; CARVALHO; CAMPELLO, 2013; ZIMEK; VREEKEN, 2015). A diversidade é vantajosa, pois considerar múltiplas partições iguais — ou quase iguais — provavelmente levaria aos mesmos problemas que acontecem quando uma única interpretação dos dados é considerada. Já a qualidade é desejada, pois agrupamentos de baixa qualidade teriam pouco a adicionar para um entendimento melhor dos dados.

Logo, para melhorar esses dois fatores — qualidade e diversidade —, e provavelmente a qualidade dos resultados, uma seleção de partições que escolha apenas aquelas que contenham esses fatores dentro de um grupo maior de agrupamentos, pode ser aplicada. Experimentalmente, em diversos casos, os resultados produzidos utilizando apenas um subconjunto selecionado de partições provou ser similar ou melhor que aqueles baseados no conjunto todo (FERN; LIN, 2008; SAKATA et al., 2010; NALDI; CARVALHO; CAMPELLO, 2013; LEI et al., 2014; LEI et al., 2016; ANTUNES; FACELI; SAKATA, 2017).

Diferentes estratégias de seleção foram avaliadas em diferentes contextos até então. Fern e Lin (2008), avaliaram, utilizando múltiplos métodos de seleção, como a qualidade e a diversidade poderiam afetar a performance de *ensembles*. Sakata et al. (2010), propuseram um método de seleção de partições como ferramenta de pós-processamento para reduzir a quantidade de resultados produzidos por algoritmos multiobjetivo. Naldi, Carvalho e Campello (2013) expandiram o trabalho de Fern e Lin (2008) avaliando mais métodos

[3] A descrição do ambiente computacional no qual os experimentos contidos nesse trabalho foram realizados, está contida no Apêndice B.

de seleção com um número maior de algoritmos de *ensemble* e *datasets*. Lei et al. (2014), Lei et al. (2016) também utilizaram as ideias de Fern e Lin (2008), mas com um rumo diferente e avaliaram como a seleção de partições afeta a área de *meta-clustering*. Por fim, Antunes, Faceli e Sakata (2017), também propuseram um método de seleção e inicialmente o compararam com os resultados do trabalho de Sakata et al. (2010). Antunes (2018) também avaliou seu método de seleção como ferramenta de pós-processamento para técnicas multiobjetivo.

Dadas essas descobertas, no presente trabalho também investigamos esses métodos de seleção. Contudo, avaliamos como esses métodos podem afetar o MOCLE. Experimentos preliminares indicam que melhorias em qualidade e reduções ainda maiores no número de partições e tempo de execução podem ser obtidas (PEDOTE; FACELI; SAKATA, 2017).

1.3 Estrutura do Trabalho

Para facilitar a leitura e compreensão desta dissertação, seus capítulos estão organizados da seguinte maneira.

- O Capítulo 2 apresenta o *framework* MOCLE e detalhes do seu funcionamento.
- O Capítulo 3 faz uma revisão da literatura dos métodos de seleção disponíveis e apresenta todos os que foram avaliados.
- O Capítulo 4 descreve os experimentos realizados, os dados utilizados e a abordagem empregada para avaliar o impacto de cada uma das seleções sobre o MOCLE.
- O Capítulo 5 apresenta e analisa os resultados experimentos realizados, demonstrando que a abordagem adotada atinge os objetivos estabelecidos.
- Finalmente, a Conclusão apresenta as considerações finais do presente trabalho e suas limitações, os principais resultados obtidos e também indicações de trabalhos futuros.

2 *Ensemble* Multiobjetivo

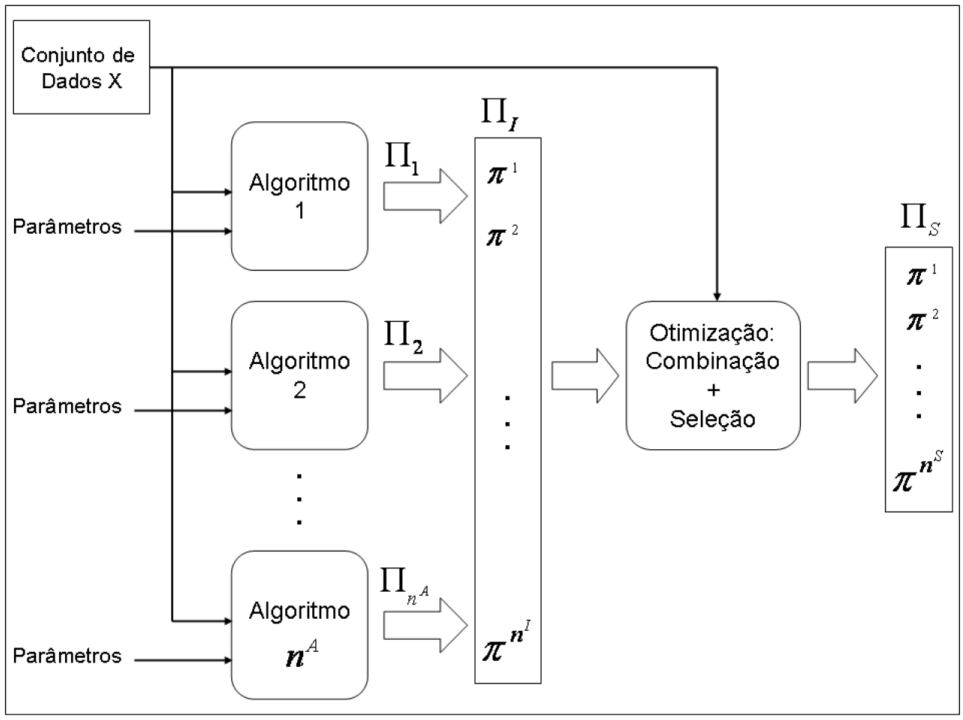
O Multi-Objective Clustering Ensemble (MOCLE), é um *framework* que realiza automaticamente diversos passos do processo de análise de agrupamentos (FACELI et al., 2009). Seu objetivo principal é amenizar conhecidos problemas relacionados a análise de agrupamento tradicional, como por exemplo: a determinação do melhor critério de agrupamento e das melhores soluções obtidas para os dados em questão. O intuito é realizar esses passos automaticamente com o menor grau de interação do usuário possível, com a menor quantidade de parâmetros. Além disso, o MOCLE, também visa corrigir algumas limitações de *ensemble* e agrupamentos multiobjetivo, através da combinação das duas técnicas. Experimentalmente, bons resultados têm sido obtidos sem a necessidade de ajustes finos em seus parâmetros (FACELI, 2006; FACELI; CARVALHO; SOUTO, 2006; FACELI; CARVALHO; SOUTO, 2007a; FACELI; CARVALHO; SOUTO, 2007b; FACELI; CARVALHO; SOUTO, 2008; FACELI et al., 2009; FACELI; SOUTO; CARVALHO, 2010; PIANTONI et al., 2015).

O MOCLE é uma das peças centrais da análise contida nesse trabalho, por isso, se faz necessário o entendimento dos pontos principais de seu funcionamento. A Figura 4 demonstra a arquitetura geral do MOCLE, que assim como técnicas de *ensemble* tem seu funcionamento divido em duas etapas. A geração das partições base é tratada na Seção 2.1, e a determinação de resultado consenso é discutida na Seção 2.2.

2.1 Geração das Partições Base

A geração do conjunto inicial de partições do MOCLE — as chamadas partições base (Π_I) — acontece de forma semelhante a *ensembles* heterogêneos (FACELI et al., 2011). Isto é, são utilizados diversos algoritmos de agrupamento conceitualmente diferentes nos dados, com múltiplas execuções utilizando diferentes parâmetros, como ilustrado na Figura 4. O intuito desse processo é obter partições que contenham diferentes vieses, critérios de agrupamentos e explorem os dados em múltiplos níveis de refinamento (i.e., diferente número de *clusters*) (FACELI; CARVALHO; SOUTO, 2006). Esses múltiplos agrupamentos são essenciais, pois como não há *a priori* nenhum conhecimento das reais estruturas presentes nos dados e também não há informações de como evidenciá-las, qualquer uma delas é potencialmente relevante.

Figura 4 – Arquitetura do MOCLE



Fonte: (FACELI, 2006)

2.2 Determinação do Consenso

Geradas as partições base, o próximo passo é a determinação de um conjunto consenso através de um algoritmo genético multiobjetivo baseado em Pareto, que é a peça central do funcionamento do MOCLE (FACELI, 2006). Esse algoritmo é um processo de otimização que, iterativamente, seleciona e combina partições (como ilustrado na Figura 4). O resultado desse processo é um conjunto de partições, que podem ser combinações de partições fornecidas ou partições de alta qualidade que aparecem no conjunto base (Π_I)(FACELI et al., 2009). Vale lembrar que para a construção de Π_S qualquer algoritmo genético multiobjetivo baseado em Pareto pode ser utilizado (FACELI, 2006).

Adicionalmente, o que confere a característica de *ensemble* ao MOCLE é seu operador especial de recombinação. Qualquer algoritmo de *ensemble*, que possa ser utilizado em pares de partições, pode ser utilizado como operador de recombinação no MOCLE (FACELI et al., 2009). Resumidamente, com esse operador especial, as partições na fronteira de Pareto são combinadas em pares, iterativamente, durante o processo de evolução do algoritmo. As partições combinadas, através do operador também são utilizadas nas próximas iterações e, possivelmente, modificam a fronteira de Pareto. Esse processo iterativo, segundo Faceli, Carvalho e Souto (2008), Faceli et al. (2011), evita a influência negativa de partições de baixa qualidade, que afeta as abordagens tradicionais de *ensemble*, pois essas são gradualmente descartadas.

Outra característica importante do MOCLE é que ele não utiliza nenhum operador de mutação, como a maioria dos algoritmos multiobjetivo (FACELI et al., 2009). Isso, acaba por gerar um conjunto de resultados mais conciso (i.e., com número menor de partições), já que uma parte menor do espaço de busca disponível é explorado.

3 Estratégias de Seleção

Nesse capítulo são apresentados os métodos de seleção de partições utilizados em nossos experimentos. Em nossa avaliação incluímos métodos de diversas abordagens de agrupamento, entre elas: *ensemble clustering*, *meta-clustering* e métodos que foram utilizados na literatura de forma independente como ferramentas de pós-processamento. Foram incluídos apenas aqueles que obtiveram melhor performance em cada trabalho na literatura disponível. No melhor de nosso entendimento, apenas o trabalho de (RASTIN; KANAWATI, 2015) foi excluído, devido a proposta estar em estágio inicial e utilizar apenas *datasets* baseados em grafos.

Todos os métodos de seleção aqui descritos funcionam recebendo um conjunto de partições, chamado de conjunto completo (Π_C) e devolvem um subconjunto do mesmo, menor que o primeiro[1], aqui chamado de conjunto reduzido (Π_R).

Exceto por dois métodos de seleção, discutidos a seguir, todos eles também recebem como parâmetro o tamanho pretendido desse subconjunto (Π_R). Alguns deles também precisam receber o *dataset* do qual essas partições se originam, pois utilizam essa informação para decidir quais partições manter e quais descartar. Porém, vale lembrar, que nenhuma dessas seleções recebe diretamente estruturas conhecidas ou criam novas partições. O funcionamento dos métodos consiste apenas em selecionar partições, segundo os critérios adotados por eles, seja através do descarte de partições que não atendem esses critérios ou pela seleção daquelas que atendem.

3.1 Diversidade

A Diversidade, como o próprio nome diz, objetiva selecionar uma aproximação do subconjunto (Π_R) mais distinto possível (FERN; LIN, 2008). O método busca uma aproximação, pois selecionar o (Π_R) mais distinto exigiria muito esforço computacional, já que esse problema é NP-*Hard* (FERN; LIN, 2008).

A Diversidade é um algoritmo guloso que inicialmente seleciona, em Π_C, a partição mais parecida com todas as outras e a adiciona no conjunto Π_R. Depois, iterativamente, são selecionadas outras partições de Π_C e adicionadas em Π_R, de forma que a diversidade de Π_R seja maximizada. Os passos desse processo são apresentados no Algoritmo 1. Lembrando que $|.|$ significa a cardinalidade de um conjunto, isto é, a cardinalidade de um conjunto qualquer Π é a quantidade de objetos (partições) que o compõe.

[1] O conjunto reduzido também pode ter tamanho igual ao conjunto completo, porém, na prática, isso não deve ocorrer ao não ser que o método de seleção seja parametrizado nesse sentido.

No algoritmo da Diversidade, para medir o quanto uma partição é parecida com a outra, e o quanto um subconjunto é diverso, um índice de validação externo é utilizado para comparar pares de partições. O índice *Normalized Mutual Information* (NMI) (STREHL; GHOSH, 2003a) é adotado pelos autores, porém, como os mesmos destacam, qualquer índice externo pode ser utilizado. No presente trabalho o *Adjusted Rand Index* (ARI) (HUBERT; ARABIE, 1985) é utilizado[2], sua descrição pode ser encontrada na Seção A.1 do Apêndice A. O índice foi alterado, nessa e em outras seleções, para manter uma padronização e evitar influências de vieses de diferentes índices. Além disso, segundo Faceli et al. (2011), o ARI é um dos índices mais utilizados para validação externa de agrupamentos.

Algoritmo 1: *Diversidade*

Entrada: Um conjunto de partições, Π_C, e o número de partições a serem selecionadas, s sendo $(s < |\Pi_C|)$

início

  $\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,max} \left\{ \sum_{\pi^j, i \neq j \in \Pi_C} ARI(\pi^i, \pi^j) \right\}$

  inserir π^t em Π_R

  remover π^t de Π_C

  repita

    $\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,min} \left\{ \sum_{\pi^j \in \Pi_R} ARI(\pi^i, \pi^j) \right\}$

    inserir π^t em Π_R

    remover π^t de Π_C

  até $|\Pi_R| = s$;

fim

retorna Π_R

3.2 *Cluster And Select* (CAS)

O *Cluster And Select* (CAS) é a seleção que mais tem impacto positivo sobre os ensembles considerados em Fern e Lin (2008). Ela, assim como a seleção Diversidade, busca um Π_R diverso, porém nessa seleção há um controle para também incluir partições de qualidade. Para isso, Fern e Lin (2008), definem qualidade de uma partição sob a ótica do quanto ela é parecida com as demais. Isso porque, se uma partição é parecida com as demais, intuitivamente, captura melhor a tendência geral do conjunto em que está inserida (FERN; LIN, 2008). A qualidade de uma partição π^i é avaliada pela função Q,

[2] Essa escolha, embora pareça contradizer o objetivo de reduzir a complexidade computacional do *framework* MOCLE, por não ser o índice com menor complexidade computacional disponível na literatura, na verdade, não o faz. Isso é observado no Capítulo 5, no qual, através de análises, fica evidente que o tempo de processamento das seleções é quase insignificante quando comparado ao tempo de processamento da fase otimização do *framework*. Logo, essa escolha não tem impacto na performance geral do MOCLE.

descrita a seguir, onde Π é um conjunto de partições qualquer no qual π^i esteja inserida.

$$Q(\pi^i, \Pi) = \sum_{\pi^i, \pi^j \in \Pi,\, i \neq j} ARI(\pi^i, \pi^j) \tag{3.1}$$

O CAS pode ser resumido em duas etapas. Primeiro, as partições disponíveis são divididas em s grupos por similaridade — partições parecidas são agrupadas em um mesmo grupo, e as diferentes em outros grupos. Depois, de cada um desses grupos, é obtida a partição com maior qualidade, isto é, aquela que é mais parecida com o restante de seu grupo, sendo essa uma representante, o mais fiel possível das características desse grupo. Essas partições de maior qualidade são aquelas que compõem Π_R.

Em maior detalhe, na primeira etapa, é construída uma matriz de similaridade entre todos os elementos de Π_C, cuja notação é dada por: $M_{p\times p}$, onde $p = |\Pi_C|$ e $M_{i,j} = ARI(\pi^i, \pi^j)$. Depois, essa matriz é particionada em s *clusters* com o uso do algoritmo *Spectral Clustering* (NG; JORDAN; WEISS, 2001). Por fim, na segunda etapa, de cada um desses *clusters* são selecionadas aquelas partições que maximizam Q, isto é, aquelas partições de maior qualidade. O funcionamento do CAS está descrito no Algoritmo 2.

Algoritmo 2: *Cluster And Select* (CAS)

Entrada: Um conjunto de partições, Π_C, e o número de partições a serem selecionadas s, sendo $(s < |\Pi_C|)$

início

$M_{p\times p} \leftarrow \begin{pmatrix} M_{1,1} & \cdots & M_{1,p} \\ M_{p,1} & \cdots & M_{p,p} \end{pmatrix}$

$\pi \leftarrow$ aplicar *Spectral Clustering* em $M_{p\times p}$ para formar uma partição com s *clusters*, $\pi = \{c_1, \ldots, c_s\}$

para todo $c_i \in \pi$ **faça**

$\pi^t \leftarrow \underset{\pi^i \in c_i}{arg\,max} Q(\pi^i, c_i)$

inserir π^t em Π_R

fim

fim

retorna Π_R

3.3 *Automatic Selection Algorithm* (ASA)

O *Automatic Selection Algorithm* (ASA), de Sakata et al. (2010), é uma ferramenta de pós-processamento para reduzir o número de agrupamentos gerados por algoritmos multiobjetivo, que, frequentemente, retornam grandes conjuntos de partições (FACELI; SOUTO; CARVALHO, 2008). O objetivo dessa redução é facilitar o processo de análise manual desses resultados. De forma similar ao método CAS, esse método de seleção

utiliza o ARI para identificar partições diversas e também para avaliar a qualidade das mesmas. O ASA define qualidade como representatividade, isto é, se uma partição tem alta similaridade com as partições do conjunto em que ela está inserida, então ela é mais fácil de ser encontrada, logo, ela é mais evidente e com maior qualidade.

O ASA consegue inferir automaticamente o tamanho do conjunto Π_R, utilizando um limiar do ARI obtido empiricamente (SAKATA et al., 2010). Por essa razão, não é possível informar diretamente ao algoritmo qual o tamanho desejado de Π_R.

Seu funcionamento é iterativo. Primeiro, as partições duplicadas que ultrapassam o número de algoritmos de agrupamento utilizados para gerar o Π_C são adicionadas a Π_R e removidas de Π_C. Depois, as partições com maior ARI médio — mais evidentes — são adicionadas iterativamente a Π_R e as partições que são similares a elas descartadas. Há também um limiar, que decrementa a cada iteração, que serve para remover de Π_C partições semelhantes as demais. O funcionamento completo do ASA está descrito nos Algoritmos 3 e 4 — a descrição por separada em duas partes devido a sua extensão.

Algoritmo 3: *Automatic Selection Algorithm* (ASA) - Parte 1

Entrada: Um conjunto de partições, Π_C, e o número de algoritmos utilizados para a geração de Π_C, p, sendo $(p < |\Pi_C|)$

início

  para todo $\pi^i \in \Pi_C$ **faça**

    Π_{ident} = partições em Π_C idênticas a π^i

    se $|\Pi_{ident}| \geq p$ **então**

      inserir π^i em Π_R

      remover π^i de Π_C

      remover Π_{ident} de Π_C

    fim

  fim

  para todo $\pi^i \in \Pi_C$ **faça**

    $ARI_m(\pi^i) = \frac{1}{|\Pi_C|} \sum_{\pi^j \in \Pi_C} ARI(\pi^i, \pi^j)$

  fim

Algoritmo 4: *Automatic Selection Algorithm* (ASA) - Parte 2

$nInicial \leftarrow |\Pi_C| + |\Pi_R|$
$t \leftarrow 0.9$ // valor inicial do limiar
$r_{atual} \leftarrow 1.0$ // razão atual de redução
$\Pi_{atual} \leftarrow \Pi_C$
repita
  $r_{anterior} \leftarrow r_{atual}$
  $\Pi_{anterior} \leftarrow \Pi_{atual}$
  $\Pi_{current} \leftarrow \emptyset$
  para todo $\pi^i \in \Pi_R$ **faça**
    para todo $\pi^j \in \Pi_C$ **faça**
      se $ARI(\pi^i, \pi^j) \geq t$ **então**
        remover π^j de Π_C
      fim
    fim
  fim
  enquanto $|\Pi_C| > 0$ **faça**
    $\pi^d \leftarrow \underset{\pi^i \in \Pi_C}{arg\,max}\, ARI_m(\pi^i)$
    remover π^d de Π_C
    inserir π^d em Π_{atual}
    para todo $\pi^i \in \Pi_C$ **faça**
      se $ARI(\pi^i, \pi^d) \geq t$ **então**
        remover π^i de Π_C
      fim
    fim
  fim
  $t \leftarrow t - 0.1$
  $\Pi_C \leftarrow \Pi_{atual}$
  $r_{atual} \leftarrow |\Pi_{atual}|/nInicial$
até $(r_{anterior} - r_{atual}) \leq 0.12$ *ou* $t < 0.1$;
inserir $\Pi_{anterior}$ em Π_R
fim
retorna Π_R

3.4 *Combination of Relative Indexes* (CRI)

Em Naldi, Carvalho e Campello (2013), assim como em Fern e Lin (2008), diversas seleções também foram avaliadas na tentativa de melhorar a performance de algoritmos de *ensemble*. A diferença, é que Naldi, Carvalho e Campello (2013), em suas seleções,

considerou diferentes índices relativos de validação de agrupamentos para medir a qualidade das partições. Seis índices foram utilizados para essas seleções, sendo que todos eles visam favorecer *clusters* bem separados e compactos (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010). A diferença entre eles se dá pela forma em que isso é calculado e em como cada medida favorece essas duas características. Todos os seis foram também utilizados no presente trabalho[3], a descrição de todos pode ser obtida em Vendramin, Campello e Hruschka (2010), são eles:

- *Dunn*;
- *Simplified Silhouette* (SS);
- *Alternative Simplified Silhouette* (ASS);
- *Calinski-Harabasz* (VRC);
- PBM; *e*
- *Davies-Bouldin* (DB).

Com o uso desses índices foram propostas seleções que os utilizam individualmente, as chamadas *Single Index Selection* (SIS) e seleções que fazem combinações desses índices, as chamadas *Combination of Relative Indexes* (CRI). Segundo Naldi, Carvalho e Campello (2013), duas categorias foram criadas, pois a abordagem dos métodos SIS de considerar apenas um índice para todas as situações, não é a melhor opção. Primeiro, porque é difícil selecionar apenas um índice para uma situação na qual — na maioria dos casos reais — não há informação disponível sobre o comportamento e performance desse índice. Segundo, porque o uso do mesmo critério para selecionar todas as partições desejadas pode gerar um conjunto reduzido enviesado, no qual as partições melhor avaliadas provavelmente terão características similares — senão iguais[4]. Então, para lidar com essas dificuldades, um comitê de índices relativos de validação (CRI) seria uma melhor alternativa, visando que a boa performance da maioria dos índices compensem o desempenho fraco de alguns (NALDI; CARVALHO; CAMPELLO, 2013). Essa estratégia de comitê, pode ser feita de várias formas, em Naldi, Carvalho e Campello (2013), foram apresentadas três, que estão listadas a seguir.

[3] Embora não ótimos para objetivos aqui dados, como será melhor visto no Capítulos 4 e 5, esses seis índices foram adotados seguindo o trabalho de Naldi, Carvalho e Campello (2013), com isso pretende-se evitar introduzir vieses indesejados. Em trabalhos futuros, como destacado nas Conclusões do presente esforço, há a possibilidade de avaliar outras combinações de índices relativos.

[4] Vale lembrar que essas observações também são válidas no contexto do presente trabalho. Pois, o impacto das seleções SIS sobre o MOCLE foi avaliado em experimentos preliminares, em Pedote, Faceli e Sakata (2017); no qual, essas seleções apresentaram desempenhos insatisfatórios e instáveis. Por esse motivo, os métodos SIS não são considerados no presente trabalho.

Contudo, antes de apresentar os métodos CRI, é preciso definir como a avaliação de cada índice é considerada no processo de escolha das melhores partições. Para esse propósito, Naldi, Carvalho e Campello (2013), definem a função *rank*, que retorna à classificação de uma dada partição π^m contida de em um dado conjunto Π quando avaliada de acordo com o u-ésimo índice relativo do comitê, aqui definido por $index_u$. Por exemplo, se a partição π^m é uma partição pertencente ao conjunto Π e é a melhor avaliada de acordo com o u-ésimo índice do comitê, então $rank(index_u, \pi^m, \Pi) = 1$. Por outro lado, se π^m é a segunda melhor partição então $rank(index_u, \pi^m, \Pi) = 2$ e assim em diante.

3.4.1 *Sum of Ranks* (SR)

O método de seleção *Sum of Ranks* (SR) funciona da seguinte forma. Primeiro, para cada índice relativo, são avaliadas e classificadas todas as partições em Π_C, da melhor partição para a pior com o uso da função *rank*. Depois, a soma das classificações de cada partição é feita e, por fim, as partições com as menores somas, isto é, aquelas melhor avaliadas por diferentes índices, são iterativamente selecionadas para compor Π_R.

Algoritmo 5: *Sum of Ranks* (SR)

Entrada: Um conjunto de partições, Π_C, o número de partições a serem selecionadas, s, sendo $(s < |\Pi_C|)$, e um conjunto de índices relativos, de tamanho v, $index_u (u = 1, \ldots, v)$;

início

  repita

    $\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,min}\, f(\pi^i) = \sum_{u=1}^{v} rank(index_u, \pi^i, \Pi_C)$

    inserir π_t em Π_R

    remover π_t de Π_C

  até $|\Pi_R| = s$;

fim

retorna Π_R

3.4.2 *Best Rank Position* (BRP)

O método BRP é similar ao SR, pois também considera todos os índices do comitê e classifica todas as partições de acordo com cada índice. Porém, em vez de uma soma das classificações, essa estratégia considera cada uma das classificações intercaladamente. Por exemplo, a primeira partição selecionada seria a melhor classificada de acordo com o primeiro índice; a segunda selecionada seria a melhor classificada de acordo com o segundo

índice, e assim por diante, até que Π_R esteja completo.

Algoritmo 6: *Best Rank Position* (BRP)

Entrada: Um conjunto de partições, Π_C, o número de partições a serem selecionadas, s, sendo $(s < |\Pi_C|)$, e um conjunto de índices relativos, de tamanho v, $index_u(u = 1, \ldots, v)$;

início

$i \leftarrow 0$

repita

$u \leftarrow u + 1$

$\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,min}\, f(\pi^i) = rank(index_u, \pi^i, \Pi_C)$

inserir π_t em Π_R

remover π_t de Π_C

$u \leftarrow u \bmod v$

até $|\Pi_R| = s$;

fim

retorna Π_R

3.4.3 *Sum of Ranks with Diversity* (SRD)

A seleção SRD, também é similar ao SR, e funciona com base no mesmo princípio de classificação das partições Π_C pelos índices relativos. Porém, em vez de apenas somar as classificações, a diversidade das mesmas também é considerada.

Para considerar a diversidade das partições a Equação 3.2 é utilizada. O princípio do cálculo é o mesmo da seleção Diversidade de Fern e Lin (2008) (descrita na Seção 3.1).

$$diversity(\pi^i, \Pi) = 1 - \sum_{\pi^j \in \Pi, i \neq j} \frac{ARI(\pi^i, \pi^j)}{|\Pi| - 1} \tag{3.2}$$

O Algoritmo 7 contém a descrição completa do funcionamento do SRD.

Algoritmo 7: *Sum of Ranks with Diversity* (SRD)

Entrada: Um conjunto de partições, Π_C, o número de partições a serem selecionadas, s, sendo $(s < |\Pi_C|)$, e um conjunto de índices relativos, de tamanho v, $index_u(u = 1, \ldots, v)$;

início

repita

$\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,min}\, f(\pi^i) = (1 - diversity(\pi^i, \Pi_C)) * \sum_{u=1}^{v} rank(index_u, \pi^i, \Pi_C)$

inserir π^t em Π_R

remover π^t de Π_C

até $|\Pi_R| = s$;

fim

retorna Π_R

3.5 *Filtered Meta-clustering* (FILTA)

Em *meta-clustering*, introduzido em Caruana et al. (2006), a intenção é agrupar um conjunto de partições gerando um conjunto de *meta-clusters* menor que o primeiro, removendo assim as partições similares. Esse esforço teve continuidade em Zhang e Li (2011) que usou *ensemble clustering* para capturar múltiplos agrupamentos a partir desses *meta-clusters*. Porém, há problemas nessas abordagens e outras similares, assim como pontuado em Muller et al. (2010), pois são suscetíveis a ruído e também falta estabilidade nos resultados, que são altamente dependentes da distribuição de qualidade e diversidade no conjunto inicial de partições. Por isso Lei et al. (2014), propôs um método de filtragem de partições antes do processo de geração desses *meta-clusters*. Em Lei et al. (2016) esse filtro recebeu modificações na sua parametrização e o algoritmo proposto para obtenção desses agrupamentos foi expandido.

Aqui, utilizamos apenas o método de seleção contido no *framework* proposto por Lei et al. (2016), o *Filtered Meta-clustering* (FILTA). O FILTA, visa obter partições que sejam de qualidade e ao mesmo tempo diversas. A diversidade das partições é obtida de forma similar a seleção Diversidade de Fern e Lin (2008). Para avaliar a qualidade das partições um estimador de entropia diferencial, o *meanNN*, é utilizado (FAIVISHEVSKY; GOLDBERGER, 2010). O *meanNN*, é utilizado nessa seleção como um índice relativo de qualidade, embora não seja reconhecido na literatura como um, seu cálculo é dado por:

$$H(\pi^m) = \sum_{j=1}^{n_c} \frac{1}{n_j - 1} \sum_{i \neq l, c_i = c_l = j} log||x_i - x_j|| \tag{3.3}$$

onde, n_c é número de *clusters* da partição π^m e x_i é o i-ésimo objeto do conjunto de dados X.

O FILTA é descrito pelo Algoritmo 8. A abordagem é incremental e semelhante a seleção Diversidade, no sentido de que, a primeira partição selecionada é a de maior qualidade e as outras são iterativamente adicionadas visando manter a diversidade de Π_R,

porém também considerando a qualidade das partições.

Algoritmo 8: *Filtered Meta-clustering* (FILTA)

Entrada: Um conjunto de partições, Π_C, o número de partições a serem selecionadas, s sendo ($s < |\Pi_C|$), e os parâmetros numéricos β e β_0 para calibrar se a qualidade ou a diversidade devem ser favorecidas (aqui, nenhuma das duas é, assim como feito pelos autores do método)

início

$\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,max}\, H(\pi^i)$

inserir π^t em Π_R

remover π^t de Π_C

repita

$\pi^t \leftarrow \underset{\pi^i \in \Pi_C}{arg\,min} \left\{\beta\beta_0 H(\pi^i) + \frac{1-\beta}{|\Pi_R|} \sum_{\pi^j \in \Pi_R} ARI(\pi^i, \pi^j)\right\}$

inserir π^t em Π_R

remover π^t de Π_C

até $|\Pi_R| = s$;

fim

retorna Π_R

3.6 *Hybrid Selection Strategy* (HSS)

O *Hybrid Selection Strategy* (HSS), como o próprio nome diz, é um algoritmo híbrido de seleção partições. É híbrido, pois combina técnicas de agrupamento multiobjetivo e de seleção de partições (ANTUNES; FACELI; SAKATA, 2017). Até então, foi testado apenas como uma ferramenta de pós-processamento, visando facilitar a análise manual de grandes conjuntos de partições; sejam eles resultantes da execução de múltiplos algoritmos tradicionais de agrupamento ou resultantes de técnicas multiobjetivo como o *Multi-Objective Clustering with automatic K-determination* (MOCK) e o MOCLE (ANTUNES, 2018).

O HSS recebe um conjunto de partições (Π_C) e tem por objetivo selecionar um subconjunto reduzido (Π_R), no qual, as partições que o compõem sejam de alta qualidade de acordo com os objetivos definidos e dissimilares entre si. Dois objetivos são adotados para avaliar a qualidade em Antunes, Faceli e Sakata (2017), Antunes (2018): variância intra-*cluster* e conectividade (ambos descritos no Apêndice A). Os mesmos objetivos são utilizados aqui.

O HSS, assim como o ASA de Sakata et al. (2010) (descrito na Seção 3.3), consegue inferir automaticamente o tamanho do conjunto Π_R. Por essa razão, não é possível informar diretamente ao algoritmo qual o tamanho desejado de Π_R.

O algoritmo pode ser dividido em três tarefas:

1. A variância intra-*cluster* e conectividade de cada partição em Π_C é calculada e depois

esses valores são usados de entrada para um algoritmo calcular a fronteira de Pareto.

2. O conjunto não dominado da fronteira, P, é particionado em *clusters* utilizando um algoritmo de agrupamento; aqui, o *K-Means* é utilizado para isso, seguindo a abordagem de Antunes (2018). Cada um desses *clusters* contém partições que correspondem a uma região da fronteira, sendo que o número de regiões, nR, é dado por uma porcentagem do número de partições na fronteira, cujo valor é fornecido através de um parâmetro. Aqui, também seguindo a abordagem das autoras da técnica, o valor do parâmetro *porcentagem* foi configurado para 30%.

3. As melhores partições de cada região são selecionadas para compor Π_R. Para identificar as melhores partições, o critério do quanto ela é parecida com as demais é utilizado. Para medir isso, o ARI médio das partições é utilizado (Equação 3.4) e aquelas que o maximizam são escolhidas para compor Π_R. Essa abordagem é semelhante ao método de seleção CAS (Equação 3.1).

$$ARI_m(\pi^j, \Pi) = \frac{1}{|\Pi|} \sum_{\pi^k \in \Pi} ARI(\pi^j, \pi^k) \tag{3.4}$$

Algoritmo 9: *Hybrid Selection Strategy* (HSS)

Entrada: Um conjunto de partições, Π_C, a porcentagem para o cálculo do número de regiões, *porcentagem*, e o conjunto de dados, X, do qual Π_C provem;

início

  para todo $\pi^i \in \Pi_C$ **faça**

    $var(\pi^i) \leftarrow variancia(\pi^i, X)$

    $con(\pi^i) \leftarrow conectividade(\pi^i, X)$

  fim

  $P \leftarrow fronteiraDePareto(var, con)$

  calcular nR, de acordo com o parâmetro *porcentagem*

  $\pi \leftarrow k\text{-}means(P, nR)$

  para todo $c_i \in \pi$ **faça**

    $\pi^m \leftarrow \overset{arg\,max}{\pi^i \in c_i} ARI_m(\pi^i, c_i)$

    inserir π^m em Π_R

  fim

fim

retorna Π_R

4 Métodos e Experimentos

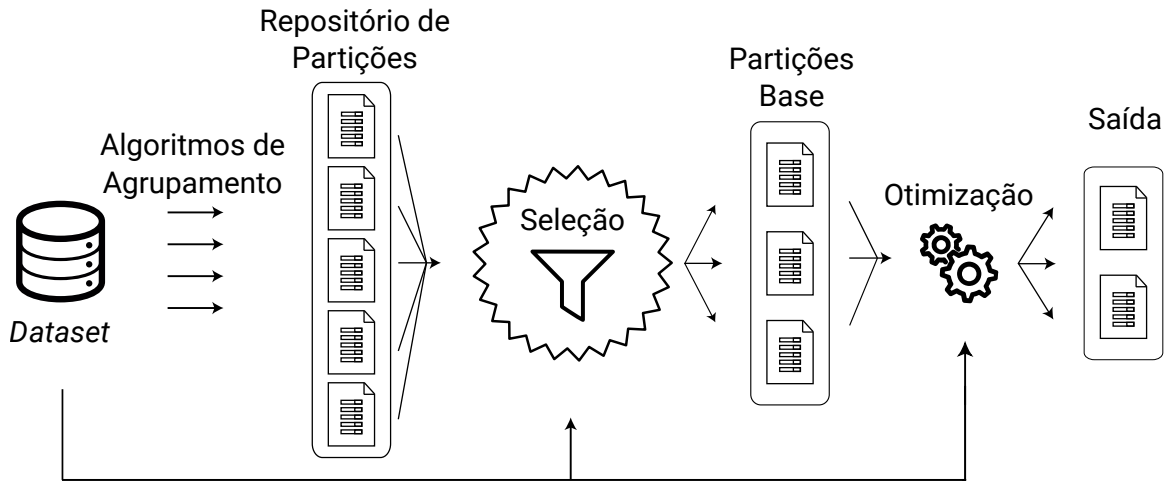
Nesse capítulo estão descritos os experimentos realizados para comparar o impacto de cada um dos algoritmos descritos no Capítulo 3 no MOCLE; também está descrita a abordagem para interpretar os resultados obtidos.

A abordagem de análise adotada nesse trabalho é baseada na aplicada em Faceli (2006), porém com objetivos diferentes. Aqui, essas análises são feitas para observar qual ou quais métodos de seleção existentes na literatura melhor se adéquam aos objetivos propostos — se é que eles existem e sob que circunstâncias tais condições são atendidas. Ademais, diferentemente de Faceli (2006), são utilizados mais *datasets* e, fora a qualidade dos resultados e a concisão deles (número de partições), também é observado o tempo de execução do *framework* MOCLE, fator importante assim como Faceli, Carvalho e Souto (2008) pontuam.

4.1 Experimentos

Para analisar o impacto das seleções nos resultados do MOCLE, foram introduzidas duas mudanças em sua arquitetura tradicional — descrita na Figura 4, na página 10. Primeiro, os algoritmos de agrupamento do *framework* passaram a gerar um repositório de partições (Π_{Rep}), em vez de gerar diretamente o conjunto de partições base (Π_I). A segunda mudança foi incluir o passo da seleção de partições, que é feita sobre Π_{Rep} e fornece o conjunto reduzido de partições (Π_R) para o MOCLE. Essas duas mudanças são ilustradas na Figura 5.

Figura 5 – Arquitetura simplificada do MOCLE, com o passo adicional de seleção.



Fonte: elaboração própria.

Essa arquitetura modificada foi a adotada para os experimentos aqui realizados, que foram conduzidos da seguinte forma: **(i)** Π_{Rep} foi gerado uma única vez para cada *dataset* descrito na Seção 4.2, de acordo com os passos descritos na Seção 4.3; **(ii)** a cada um dos métodos de seleção, descritos no Capítulo 3, foi fornecido o mesmo repositório de partições (Π_{Rep}); **(iii)** os resultados de cada seleção (Π_R) foram fornecidos como partições base (Π_I) para o processo de otimização.

Contudo, devido à natureza não-determinística do processo de otimização, o último passo foi executado 30 vezes para cada entrada, com as mesmas configurações iniciais. Isso resultou em 30 diferentes conjuntos de partições (Π_S) para cada método de seleção. Foram utilizadas 30 execuções, pois todos os trabalhos anteriores relacionados ao MOCLE, o fizeram. Adicionalmente, para servir de *baseline* para a avaliação do impacto das seleções, o processo de otimização foi executado 30 vezes utilizando todo o repositório de partições, isto é, seguindo a arquitetura tradicional do MOCLE, sem seleção de partições.

Para todas as seleções nas quais é possível informar o tamanho de Π_R via parâmetro (SRD, SR, BRP, Diversidade, CAS e FILTA), a redução foi empiricamente configurada para 50%[1]. Isto é, para essas seleções, Π_{Rep} foi reduzido pela metade e depois fornecido ao MOCLE. Nas outras duas (ASA e HSS), onde isso não é possível, foram adotados seus parâmetros padrão.

4.2 Conjuntos de Dados

Para analisar o impacto da seleção de partições no MOCLE foram utilizados 30 conjuntos de dados, entre eles, reais, artificiais e *datasets* de *benchmark*. Esses *datasets* possuem diferentes características: número de estruturas conhecidas (sendo representadas por $|\Pi_E|$ e o conjunto por $\Pi_E = \{\pi^{E1}, \pi^{E2}, \ldots, \pi^{En^E}\}$), número de objetos ($n$) e atributos ($d$), diferentes tipos de *clusters* (em alguns casos, no mesmo *dataset*) e diferentes níveis de refinamento de *clusters*. O intuito disso foi observar o comportamento das seleções sobre MOCLE em diferentes cenários. A Tabela 1 lista os *datasets* utilizados e suas principais características.

Todos os *datasets* aqui utilizados estão publicamente disponíveis. Os conjuntos de dados `aggregation`, `compound`, `pathbased`, `R15`, `jain` e `flame`, estão disponíveis em <http://cs.joensuu.fi/sipu/datasets/> (FRÄNTI, 2015). E todos os outros podem ser acessados em <http://lasid.sor.ufscar.br/clustersEvaluationBenchmark/> (FACELI; SAKATA, 2016).

[1] Foram realizados experimentos preliminares para reduções de 90%, 80%, 70% e 60% em Π_{Rep}, porém os melhores resultados, em termos de qualidade, para a maioria das seleções, foram obtidos com 50% de redução.

Tabela 1 – Características dos *datasets*, sendo n o número de objetos, d o número de atributos, $|\Pi_E|$ o número de estruturas conhecidas e K o número de *clusters* contidos em cada uma das estruturas conhecidas.

Tipo	*Dataset*	n	d	$\|\Pi_E\|$	K
Artificial	atom	800	3	1	2
	2sp2glob	2000	2	1	4
	aggregation	788	2	1	7
	chainlink	1000	3	1	2
	compound	399	2	1	6
	ds2c2sc13	588	2	3	2,5,13
	flame	240	2	1	2
	gaussian3	60	600	1	3
	hepta	212	3	1	7
	jain	373	2	1	2
	lsun	400	2	1	3
	monkey	4000	2	4	8,5,3,2
	pathbased	300	2	1	3
	R15	600	2	1	15
	simulated6	60	600	1	6
	spiralsquare	2000	2	2	2,6
	target	770	2	1	6
	tetra	400	3	1	4
	twoDiamonds	800	2	1	2
	wingNut	1016	2	1	2
Benchmark	glass	214	9	3	2, 5, 6
	iris	150	4	1	3
Real	chowdary	104	182	1	2
	dyrskjot	40	1203	1	3
	eTongueSugar	375	6	2	2,3
	golub	72	3571	4	2, 3, 2, 4
	gordon	181	1626	1	2
	leukemia	327	271	2	3, 7
	lung	197	1000	1	4
	miRNAcancer	218	217	6	3, 20, 4, 9, 2, 2

4.3 Geração do Repositório de Partições

O processo de geração de partições também foi similar ao adotado em Faceli (2006). Porém dois novos algoritmos de agrupamento foram utilizados, sendo eles, o *Centroid-Link* (CE) e *Complete-Link* (CO). Essa mudança foi feita para aumentar e diversificar o número de partições contido no repositório de cada *dataset*. Para os diferentes níveis de refinamento das partições contidas em cada Π_{Rep}, adotamos o intervalo $[K^{min}, K^{max}]$, com $K^{min} = 2$ e $K^{max} = 2 * K^{Ej}$, sendo π^{Ej} a estrutura com maior número de *clusters*. Isto é, todas as partições foram geradas de forma que o número de *clusters* em cada uma respeitasse esse intervalo. A seguir está detalhado como cada algoritmo tradicional foi aplicado:

***K-means* (KM):** Para evitar ótimos locais, causados por inicializações ruins, o algoritmo foi executado 30 vezes para cada valor de K — os centroides iniciais foram definidos aleatoriamente a cada execução. Dessas 30 partições, para cada valor de K, apenas aquela com menor erro quadrático foi utilizada para compor o Π_{Rep}.

***Single-Link* (SL), *Average-Link* (AL), *Centroid-Link* (CE) e *Complete-Link* (CO):** Cada uma das árvores, gerada por cada um dos algoritmos, foi cortada de modo a

obter uma partição para cada um dos valores de K.

***Shared Nearest Neighbors* (SNN):** Seguindo as observações de Faceli (2006), onde outras variações de parâmetros não surtiram efeitos relevantes, esse algoritmo foi executado com diferentes configurações de parâmetros:

- *Nearest Neighbors* - número de vizinhos considerados: 2%, 5%, 10%, 20%, 30% e 40% de n;
- *merge* - porcentagem de ligações usadas para a junção de *clusters*: 0, 20%, 40%, 60%, 80% e 100%;
- *topic* - porcentagem de pontos representativos: 0, 20%, 40%, 60%, 80% e 100%;
- *strong* - número de ligações fracas (relacionado ao peso das conexões do grafo dos vizinhos mais próximos): 0,25 (valor padrão); e
- *noise* e *label*: 0 (isto é, todos os pontos foram rotulados e nenhum foi excluído como ruído).

Dentre as partições produzidas com todas as diferentes combinações dessas variações de parâmetros, foram selecionadas para compor Π_{Rep} apenas aquelas que continham o número de *clusters* no intervalo de interesse.

Vale ressaltar que essas configurações implicam que para cada um dos algoritmos KM, SL, AL, CE, e CO, existe uma partição correspondente para cada valor de $K \in [K^{min}, K^{max}]$. Porém, para o algoritmo SNN, pode haver mais de uma partição para cada K. Dessa forma, para cada conjunto de dados, mesmo aqueles com intervalo igual de $[K^{min}, K^{max}]$, é possível que um número diferente de partições tenha sido gerado. O número de indivíduos no Π_{Rep} de cada caso pode ser observado na Tabela 2.

Para a execução de todos os algoritmos de agrupamento tradicionais citados, exceto para o SNN[2], foi utilizado o *software* Cluster 3.0[3]. Para todas as execuções do MOCLE[4], foram utilizados os parâmetros padrão. Isto é:

- O intervalo de $[K^{min}, K^{max}]$, também necessário para o MOCLE, foi o mesmo adotado para a geração dos repositórios de partições. Esses valores podem ser consultados na Tabela 2.
- O número de gerações foi fixado em 100.
- O algoritmo do operador de recombinação foi o *Meta-CLustering Algorithm* (MCLA) (STREHL; GHOSH, 2003b).

[2] Para o SNN, a implementação utilizada foi a mesma de Faceli et al. (2010).
[3] Disponível em: <http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm>.
[4] Disponível em: <http://lasid.sor.ufscar.br/mocleproject/>. Mais detalhes sobre a implementação do MOCLE estão disponíveis no Apêndice B.

- A conectividade (Apêndice A.2.8) e a variância intra-*cluster* (Apêndice A.2.7) foram utilizadas como funções objetivo.

- O *Non-dominated Sorting Genetic Algorithm* (NSGA-II) foi utilizado como algoritmo genético (FACELI, 2006).

- Para o número de vizinhos mais próximos (NN), necessário para o cálculo da conectividade, foi utilizado 5% de n (o número de objetos do *dataset*). Esses valores podem ser consultados na Tabela 2.

Tabela 2 – Valores dos parâmetros utilizados para o MOCLE.

Dataset	K^{min}	K^{max}	NN	$\lvert\Pi_{Rep}\rvert$
atom	2	4	40	21
2sp2glob	2	8	100	54
aggregation	2	14	40	74
chainlink	2	4	50	27
chowdary	2	4	6	15
compound	2	12	20	62
ds2c2sc13	2	26	30	147
dyrskjot	2	6	2	25
eTongueSugar	2	6	19	25
flame	2	4	12	25
gaussian3	2	6	3	45
glass	2	12	11	55
golub	2	8	4	49
gordon	2	4	10	15
hepta	2	7	11	88
iris	2	6	8	37
jain	2	2	19	27
leukemia	2	14	17	66
lsun	2	6	20	29
lung	2	8	10	40
miRNAcancer	2	40	11	218
monkey	2	16	200	76
pathbased	2	6	15	29
R15	2	30	30	154
simulated6	2	12	3	58
spiralsquare	2	12	100	85
target	2	12	39	55
tetra	2	8	20	53
twoDiamonds	2	4	40	24
wingNut	2	4	51	17

4.4 Metodologia de Avaliação dos Experimentos

Os resultados e execuções do MOCLE, associados a cada um dos métodos de seleção, foram analisados e comparados com o *baseline*. Para essa comparação foram utilizados os três critérios, descritos no Capítulo 1, que motivam o presente trabalho: impacto na qualidade das partições, número final de partições produzidas e impacto no tempo de execução. As seções a seguir detalham como cada desses critérios foi mensurado e avaliado.

4.4.1 Impacto na Qualidade

Para aferir a influência das diferentes seleções nos resultados do MOCLE, sua habilidade em identificar cada uma das estruturas conhecidas dos *datasets* foi mensurada. Vale notar que as estruturas conhecidas não foram utilizadas em nenhum momento anterior, seja durante a geração das partições, nas seleções ou no MOCLE. O propósito das estruturas conhecidas foi de apenas aferir a qualidade final dos resultados.

Para medir a qualidade de relacionada aos resultados de cada uma das seleções foi adotado o mesmo procedimento de Faceli (2006), descrito a seguir.

1. Primeiro, o ARI (Apêndice A, Seção A.1.1) é utilizado para identificar as partições mais parecidas com cada uma das estruturas reais (π^{Ej}), em cada um dos 30 conjuntos Π_S, relacionados a aplicação de uma seleção em um *dataset*[5]. O que implica em identificar, pelo menos, uma partição para cada um dos 30 conjuntos Π_S, já que em todos os *datasets* há, pelo menos, uma estrutura real.

2. Em seguida, é calculado a média e o desvio padrão do ARI dessas partições identificadas de acordo com cada π^{Ej} mais parecida. Isso permite sumarizar o impacto de cada seleção no MOCLE, por *dataset*, medindo a qualidade — e o quanto ela varia — de acordo com cada estrutura conhecida. O mesmo procedimento foi adotado para as 30 execuções do *baseline*.

Sumarizado o impacto de cada seleção na qualidade dos resultados, se fez necessária uma forma de identificar se houve perdas ou ganhos em relação ao *baseline*. Para tal, dois testes estatísticos, também utilizados em Faceli (2006), foram adotados. São eles, o teste de Friedman e o pós teste de Nemenyi, que segundo Demšar (2006), são apropriados para comparar vários algoritmos aplicados em múltiplos *datasets*, quando não há garantias que as suposições do teste *Analysis of variance* (ANOVA) não sejam violadas. Para a aplicação dos testes foi adotado o nível de 95% de significância estatística, também adotada em Faceli (2006). A aplicação dos testes foi feita da seguinte forma:

[5] Para evitar que haja viés nas análises feitas, já que o ARI também foi utilizado nas seleções. O resultados analisados com o ARI também foram analisados com os índices NMI e *Adjusted Mutual Information* (AMI) (SOUTO et al., 2012). Os valores das comparações foram diferentes, porém as conclusões obtidas e observações feitas no Capítulo 5 não se alteraram.

1. Para todas as seleções, e também para o *baseline*, os valores do ARI das melhores partições identificadas em relação a cada estrutura conhecida, calculados como descrito acima, foram fornecidos ao teste de Friedman. Por exemplo, para uma estrutura conhecida de um dado *dataset*, nove conjuntos[6] de 30 valores ARI foram fornecidas ao teste de Friedman;

2. Já a aplicação do pós-teste de Nemenyi se deu apenas quando hipótese nula do teste de Friedman foi rejeitada. Nesses casos, o teste Nemenyi foi utilizado para identificar quais dos oito conjuntos de 30 valores ARI eram diferentes do *baseline*.

Além disso, para facilitar a interpretação dos resultados dos testes e dos valores (das médias e desvios padrão), eles foram classificados em quatro categorias:

Estatisticamente similares: Aqueles resultados nos quais a aplicação dos dois testes estatísticos resultou na constatação de que as hipóteses nulas de ambos não foram rejeitadas. Isto é, quando ambos os testes indicaram que os valores eram estatisticamente similares.

Melhoras relevantes: Aqueles resultados em que a hipótese nula de ambos os testes foi rejeitada e houve um incremento de, pelo menos, 0,1 na média[7].

Pioras relevantes: Foram considerados aqueles resultados também estatisticamente diferentes do *baseline* (i.e., a hipótese nula foi rejeitada para ambos os testes) e onde a média diminuiu em, pelo menos, $-0{,}1$.

Variações não relevantes: Os resultados onde a média foi considerada estatisticamente diferente do *baseline*, porém a média variou menos de 0,1, para mais ou para menos.

Por fim, também foi conduzida uma análise das melhores partições selecionadas pelos métodos. Para tal, a melhor partição selecionada de cada seleção, em relação a cada estrutura conhecida, foi identificada com o uso do ARI. Essa informação foi utilizada para ajudar a verificar como a qualidade das partições fornecidas ao MOCLE influência na qualidade de seus resultados.

4.4.2 Impacto no Número Final de Partições

Outro fator analisado foi o impacto no número final de partições, já que umas das vantagens do MOCLE frente a abordagens de agrupamento puramente multiobjetivo é sua

[6] São nove conjuntos, pois são oito são relacionados aos métodos de seleção (SRD, SR, BRP, ASA, Diversidade, CAS, FILTA e HSS) e mais um que é o *baseline*.

[7] Vale lembrar que esse valor foi escolhido apenas para auxiliar no entendimento dos resultados e não tem nenhum real significado atrelado ao ARI.

tendência em apresentar resultados concisos (i.e., com menor número de partições) (FACELI et al., 2009). Além disso, em estudos preliminares, reduzir o número de partições de entrada do MOCLE acabou por também reduzir o número de saídas (PEDOTE; FACELI; SAKATA, 2017). Um conjunto de partições resultantes conciso é importante, principalmente quando a qualidade dos resultados não piora, pois facilita a análise manual por um especialista de domínio (FACELI, 2006).

Para facilitar a análise dos resultados, as mesmas quatro categorias definidas na Seção 4.4.1 foram adotadas. Porém, em vez de considerar o limiar de 0,1, para mais ou para menos, como determinante para identificar melhoras ou pioras, foi utilizado o valor 1. Isto é, houve melhora ou piora, se as médias do número de agrupamentos contidos nos resultados do MOCLE variam para uma partição a mais ou a menos.

4.4.3 Impacto no Tempo

Não há em outros trabalhos sobre o MOCLE, uma análise sobre seu tempo de execução. Porém, esse é um fator limitante em sua aplicação (FACELI; CARVALHO; SOUTO, 2008; FACELI et al., 2009). Inclusive, durante a realização dos experimentos contidos nesse trabalho, *datasets* como D31 (VEENMAN; REINDERS; BACKER, 2002), mesmo sendo relativamente pequenos[8], foram excluídos da análise devido ao tempo excessivo de processamento[9]. Além disso, experimentos preliminares demonstraram que a seleção de partições pode reduzir de forma significativa o tempo de execução do MOCLE (PEDOTE; FACELI; SAKATA, 2017).

O tempo de execução foi medido em segundos e compreende apenas a execução da fase de otimização do MOCLE. A fase de geração de partições foi ignorada, pois foi executada apenas uma única vez para cada conjunto de dados; para mais, essa fase foi executada antes da seleção, não podendo ser impactada por ela.

Para facilitar a comparação, o tempo foi calculado através da média das 30 execuções por método de seleção, individualmente para cada *dataset*. E as mesmas quatro categorias definidas na Seção 4.4.1 foram adotadas. Porém, como determinante para identificar melhoras ou pioras foi utilizado 10% das médias de cada *dataset* no *baseline*. Isso porque os valores observados variaram muito e definir um limiar fixo seria tendencioso e injusto.

O tempo de execução dos métodos de seleção também foi contabilizado, ele é apresentado de forma separada no Capítulo 5. Para essas medições, nenhum teste estatístico foi feito, dado que houve apenas uma execução por combinação de base de dados e seleção.

Um único ambiente computacional foi adotado para todas as execuções e experimentos. Sua descrição e outros detalhes de implementação estão no Apêndice B.

[8] O *dataset* D31 possuí 3100 objetos e 2 atributos.

[9] O D31, utilizando a mesma metodologia experimental aplicada nos outros *datasets* e o mesmo ambiente de execução, sem execuções paralelas, demandaria em torno de 2600 horas de processamento.

5 Resultados

Nesse capítulo serão apresentados e discutidos os resultados obtidos dos experimentos propostos no Capítulo 4. A análise dos três critérios de avaliação propostos no capítulo anterior será feita em duas partes. Primeiro, o impacto na qualidade dos resultados será discutido em conjunto com o impacto no número resultante de partições do MOCLE. Depois, será avaliado o impacto no tempo de execução do MOCLE.

5.1 Impacto na Qualidade e Número de Partições

Como discutido no Capítulo 4, para cada seleção, a qualidade das partições selecionadas e o impacto das mesmas dos resultados finais do MOCLE foi mensurado. Para auxiliar na análise, esses valores foram separados em categorias, assim como também descrito no Capítulo 4. Na Tabela 3 estão destacadas essas quatro categorias da seguinte forma. Valores destacados em **negrito** são estatisticamente similares. Os destacados em $\bigtriangledown$**vermelho** indicam pioras relevantes. Aqueles em $\bigtriangleup$**azul** indicam melhoras relevantes. E, por fim, valores não destacados indicam variações não relevantes. Na Tabela 4 estão os valores relacionados a qualidade das partições selecionadas, onde a mesma categorização e padrão de cores são adotados, porém como os valores listados nessa tabela correspondem a apenas uma observação — o ARI da melhor partição selecionada em relação a cada estrutura conhecida —, por esse motivo nenhum teste estatístico foi adotado, apenas a variação dos valores foi considerada na categorização.

Os valores relacionados ao impacto das seleções no número final de partições produzidas pelo MOCLE estão contidos na Tabela 5. Nessa tabela também estão destacadas as mesmas quatro categorias da Tabela 3, porém dois símbolos são trocados, já que reduções nos números de partições são desejáveis. Isto é, os valores destacados em $\bigtriangledown$**azul** indicam melhoras relevantes. Já aqueles em $\bigtriangleup$**vermelho** indicam pioras relevantes.

Tabela 3 – Médias e desvios padrão do ARI das melhores partições resultantes das múltiplas execuções do MOCLE sob a influência das diferentes seleções.

Dataset	Estrutura	Π_{Rep}	*Baseline*	SRD	SR	BRP	ASA	Diversidade	CAS	FILTA	HSS
atom	E1 = K2	1	1.00 ± 0.00	▽**0.45 ± 0.00**	▽**0.45 ± 0.00**	▽**0.45 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
2sp2glob	E1 = K4	1	0.71 ± 0.00	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**
aggregation	E1 = K7	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**
chainlink	E1 = K2	1	1.00 ± 0.00	▽**0.31 ± 0.00**	**1.00 ± 0.00**	▽**0.31 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
compound	E1 = K6	0.85	0.81 ± 0.03	**0.81 ± 0.01**	0.85 ± 0.00	**0.80 ± 0.01**	**0.81 ± 0.03**	0.78 ± 0.01	**0.80 ± 0.01**	**0.81 ± 0.02**	**0.82 ± 0.03**
ds2c2sc13	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	E2 = K5	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	0.96 ± 0.04
	E3 = K13	1	0.68 ± 0.04	0.65 ± 0.00	0.65 ± 0.00	0.65 ± 0.00	0.76 ± 0.00	**0.67 ± 0.05**	0.75 ± 0.05	0.65 ± 0.00	0.78 ± 0.01
flame	E1 = K2	0.94	0.97 ± 0.01	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.97 ± 0.00**	**0.97 ± 0.01**	0.93 ± 0.00	0.89 ± 0.11	**0.97 ± 0.01**	**0.96 ± 0.05**
gaussian3	E1 = K3	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
hepta	E1 = K7	0.97	0.97 ± 0.00	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**
jain	E1 = K2	1	1.00 ± 0.00	▽**0.51 ± 0.00**	**1.00 ± 0.00**	▽**0.51 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
lsun	E1 = K3	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.98 ± 0.05**	**1.00 ± 0.00**	**1.00 ± 0.01**	**1.00 ± 0.00**	**1.00 ± 0.00**
monkey	E1 = K2	0.73	0.55 ± 0.05	**0.55 ± 0.02**	**0.54 ± 0.02**	0.59 ± 0.05	0.60 ± 0.11	**0.59 ± 0.06**	**0.55 ± 0.03**	**0.50 ± 0.12**	0.56 ± 0.02
	E2 = K3	0.83	0.69 ± 0.04	**0.70 ± 0.02**	**0.70 ± 0.02**	0.70 ± 0.04	**0.68 ± 0.02**	▽**0.56 ± 0.08**	**0.70 ± 0.02**	**0.68 ± 0.04**	0.70 ± 0.01
	E3 = K5	0.65	0.57 ± 0.02	0.58 ± 0.01	0.58 ± 0.01	0.57 ± 0.03	**0.59 ± 0.02**	0.53 ± 0.04	**0.59 ± 0.02**	**0.57 ± 0.03**	**0.58 ± 0.00**
	E4 = K8	0.67	0.66 ± 0.02	0.64 ± 0.01	0.65 ± 0.01	0.65 ± 0.01	0.65 ± 0.00	**0.66 ± 0.02**	**0.67 ± 0.00**	**0.66 ± 0.01**	0.61 ± 0.00
pathbased	E1 = K3	0.49	0.49 ± 0.00	0.46 ± 0.00	**0.49 ± 0.00**	0.47 ± 0.00	0.47 ± 0.01	0.46 ± 0.01	**0.49 ± 0.00**	**0.49 ± 0.00**	**0.49 ± 0.00**
R15	E1 = K15	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**
simulated6	E1 = K6	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**
spiralsquare	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	E2 = K6	0.67	0.67 ± 0.00	**0.67 ± 0.00**	**0.67 ± 0.00**	**0.67 ± 0.00**	**0.67 ± 0.00**	0.64 ± 0.03	0.66 ± 0.00	**0.67 ± 0.00**	**0.67 ± 0.00**
target	E1 = K6	1	0.97 ± 0.00	▽**0.70 ± 0.10**	▽**0.72 ± 0.05**	**0.96 ± 0.05**	**0.93 ± 0.10**	**0.92 ± 0.11**	**0.97 ± 0.00**	**0.97 ± 0.00**	0.88 ± 0.13
tetra	E1 = K4	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
twoDiamonds	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
wingNut	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
glass	E1 = K2	0.67	0.67 ± 0.01	0.60 ± 0.00	0.60 ± 0.01	0.61 ± 0.01	**0.67 ± 0.00**	**0.67 ± 0.01**	**0.67 ± 0.01**	0.61 ± 0.01	0.62 ± 0.02
	E2 = K5	0.56	0.56 ± 0.01	▽**0.45 ± 0.00**	▽**0.45 ± 0.00**	▽**0.46 ± 0.01**	0.55 ± 0.00	0.55 ± 0.01	**0.56 ± 0.01**	0.49 ± 0.00	0.49 ± 0.03
	E3 = K6	0.26	0.26 ± 0.01	0.20 ± 0.00	0.20 ± 0.00	0.21 ± 0.01	**0.26 ± 0.00**	**0.26 ± 0.01**	**0.26 ± 0.01**	0.22 ± 0.00	0.22 ± 0.02
iris	E1 = K3	0.82	0.76 ± 0.00	**0.76 ± 0.00**	**0.76 ± 0.00**	**0.76 ± 0.00**	**0.76 ± 0.00**	**0.76 ± 0.01**	**0.76 ± 0.00**	**0.76 ± 0.00**	**0.76 ± 0.00**
chowdary	E1 = K2	0.07	0.07 ± 0.00	**0.07 ± 0.00**	0.06 ± 0.00	**0.07 ± 0.00**	0.06 ± 0.00	**0.07 ± 0.00**	0.06 ± 0.00	0.06 ± 0.00	**0.07 ± 0.00**
dyrskjot	E1 = K3	0.55	0.54 ± 0.00	**0.54 ± 0.00**	▽**0.44 ± 0.04**	0.46 ± 0.01	▽**0.38 ± 0.00**	0.49 ± 0.00	**0.54 ± 0.00**	**0.54 ± 0.00**	**0.54 ± 0.00**
eTongueSugar	E1 = K2	0.7	0.72 ± 0.11	**0.74 ± 0.13**	**0.72 ± 0.12**	**0.73 ± 0.11**	**0.62 ± 0.05**	**0.73 ± 0.13**	**0.70 ± 0.08**	**0.70 ± 0.10**	▽**0.57 ± 0.07**
	E2 = K3	0.05	0.07 ± 0.01	**0.07 ± 0.01**	**0.07 ± 0.01**	**0.07 ± 0.01**	0.05 ± 0.01	**0.07 ± 0.01**	**0.07 ± 0.01**	**0.07 ± 0.01**	0.03 ± 0.00
golub	E1 = K2	0.94	0.88 ± 0.00	0.96 ± 0.04	**0.88 ± 0.00**	**0.88 ± 0.01**	**0.88 ± 0.00**	▽**0.26 ± 0.03**	**0.88 ± 0.00**	**0.88 ± 0.00**	▽**0.63 ± 0.02**
	E2 = K2	0.31	0.31 ± 0.00	▽**0.06 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**
	E3 = K3	0.87	0.87 ± 0.01	**0.88 ± 0.02**	**0.88 ± 0.02**	▽**0.65 ± 0.01**	0.80 ± 0.00	▽**0.41 ± 0.04**	0.80 ± 0.00	**0.87 ± 0.01**	**0.88 ± 0.01**
	E4 = K4	0.73	0.67 ± 0.00	0.72 ± 0.03	**0.67 ± 0.00**	**0.67 ± 0.00**	**0.67 ± 0.00**	▽**0.23 ± 0.01**	**0.68 ± 0.01**	**0.67 ± 0.00**	▽**0.46 ± 0.00**
gordon	E1 = K2	0.17	0.22 ± 0.13	**0.22 ± 0.10**	**0.20 ± 0.09**	**0.20 ± 0.09**	△**0.39 ± 0.13**	**0.17 ± 0.00**	**0.32 ± 0.14**	**0.17 ± 0.00**	**0.17 ± 0.00**
leukemia	E1 = K3	0.33	0.41 ± 0.00	0.35 ± 0.03	▽**0.30 ± 0.02**	**0.41 ± 0.00**	**0.39 ± 0.05**	**0.41 ± 0.00**	0.33 ± 0.05	**0.41 ± 0.00**	▽**0.30 ± 0.01**
	E2 = K7	0.77	0.77 ± 0.00	▽**0.22 ± 0.00**	0.76 ± 0.00	**0.77 ± 0.00**	0.75 ± 0.00	0.72 ± 0.01	**0.77 ± 0.00**	**0.77 ± 0.00**	0.75 ± 0.00
lung	E1 = K4	0.64	0.77 ± 0.06	▽**0.64 ± 0.01**	▽**0.56 ± 0.04**	▽**0.51 ± 0.09**	**0.73 ± 0.09**	▽**0.59 ± 0.13**	0.72 ± 0.08	0.68 ± 0.09	▽**0.58 ± 0.08**
miRNAcancer	E1 = K2	0.59	0.62 ± 0.01	**0.62 ± 0.00**	**0.62 ± 0.01**	**0.62 ± 0.00**	**0.63 ± 0.05**	**0.61 ± 0.01**	**0.62 ± 0.01**	**0.61 ± 0.01**	0.60 ± 0.03
	E2 = K2	0.22	0.10 ± 0.01	**0.10 ± 0.01**	**0.10 ± 0.01**	0.09 ± 0.01	**0.11 ± 0.03**	0.12 ± 0.02	0.09 ± 0.01	**0.10 ± 0.01**	0.09 ± 0.01
	E3 = K3	0.3	0.23 ± 0.01	**0.23 ± 0.01**	**0.23 ± 0.01**	**0.23 ± 0.01**	0.22 ± 0.01	**0.23 ± 0.01**	**0.24 ± 0.01**	0.24 ± 0.01	**0.23 ± 0.00**
	E4 = K4	0.66	0.61 ± 0.03	**0.63 ± 0.03**	0.63 ± 0.03	**0.62 ± 0.03**	**0.61 ± 0.04**	**0.61 ± 0.03**	**0.62 ± 0.03**	**0.61 ± 0.04**	**0.60 ± 0.03**
	E5 = K9	0.3	0.32 ± 0.05	**0.34 ± 0.04**	**0.34 ± 0.04**	**0.33 ± 0.04**	0.28 ± 0.04	**0.29 ± 0.04**	**0.32 ± 0.05**	**0.32 ± 0.06**	0.28 ± 0.04
	E6 = K20	0.42	0.42 ± 0.00	**0.42 ± 0.00**	**0.42 ± 0.00**	**0.42 ± 0.00**	0.40 ± 0.00	0.40 ± 0.00	**0.42 ± 0.00**	**0.42 ± 0.00**	0.40 ± 0.00
Média		0.72	0.70	0.63	0.67	0.65	0.69	0.65	0.70	0.69	0.67
Perda média de qualidade em relação ao *Baseline*				9.31%	4.30%	7.61%	0.84%	6.52%	0.35%	1.23%	3.92%

Tabela 4 – ARI das melhores partições selecionadas pelos diferentes métodos de seleção.

Dataset	Estrutura	Π_{Rep}	SRD	SR	BRP	ASA	Diversidade	CAS	FILTA	HSS
atom	E1 = K2	1	▽**0,45**	▽**0,45**	▽**0,45**	1	1	1	1	1
2sp2glob	E1 = K4	1	▽**0,71**	▽**0,71**	▽**0,71**	1	1	▽**0,71**	1	1
aggregation	E1 = K7	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
chainlink	E1 = K2	1	▽**0,40**	1	▽**0,40**	1	1	1	1	1
compound	E1 = K6	0,85	0,84	0,85	0,82	0,85	0,85	0,83	0,85	0,85
ds2c2sc13	E1 = K2	1	1	1	1	1	1	▽**0,68**	1	1
	E2 = K5	1	1	1	1	0,95	▽**0,87**	1	1	0,92
	E3 = K13	1	▽**0,62**	▽**0,62**	▽**0,63**	▽**0,83**	1	1	1	▽**0,87**
flame	E1 = K2	0,94	0,94	0,94	▽**0,71**	0,94	0,92	▽**0,71**	0,94	▽**0,71**
gaussian3	E1 = K3	1	1	1	1	1	1	1	0,98	1
hepta	E1 = K7	0,97	0,97	0,97	0,97	0,97	0,97	0,97	0,97	0,97
jain	E1 = K2	1	▽**0,78**	1	▽**0,78**	1	1	1	1	1
lsun	E1 = K3	1	▽**0,68**	1	0,99	0,95	0,99	0,95	1	0,99
monkey	E1 = K2	0,73	0,68	0,68	0,73	0,63	0,68	0,73	▽**0,40**	▽**0,55**
	E2 = K3	0,83	▽**0,71**	▽**0,71**	0,83	▽**0,67**	0,83	0,82	▽**0,71**	▽**0,70**
	E3 = K5	0,65	0,58	0,58	0,65	0,58	0,65	0,65	0,58	0,58
	E4 = K8	0,67	0,65	0,65	0,65	0,65	0,67	0,67	0,67	0,61
pathbased	E1 = K3	0,49	0,46	0,49	0,47	0,45	0,45	0,49	0,49	0,49
R15	E1 = K15	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,97
simulated6	E1 = K6	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
spiralsquare	E1 = K2	1	1	1	1	1	1	1	▽**0,75**	1
	E2 = K6	0,67	0,67	0,67	0,67	0,66	0,63	0,66	0,67	0,67
target	E1 = K6	1	▽**0,63**	▽**0,67**	1	1	1	1	▽**0,82**	▽**0,66**
tetra	E1 = K4	1	1	1	1	1	1	1	1	1
twoDiamonds	E1 = K2	1	1	1	1	1	1	1	1	1
wingNut	E1 = K2	1	1	1	1	1	1	1	▽**0,86**	1
glass	E1 = K2	0,67	0,6	0,65	0,65	0,67	0,67	0,67	0,61	0,6
	E2 = K5	0,56	▽**0,45**	0,5	0,5	0,55	0,55	0,56	0,49	▽**0,45**
	E3 = K6	0,26	0,2	0,24	0,24	0,26	0,26	0,26	0,22	0,2
iris	E1 = K3	0,82	0,76	0,76	0,76	0,76	0,82	0,76	0,82	0,76
chowdary	E1 = K2	0,07	0,07	0,06	0,07	0,06	0,07	0,06	0,06	0,07
dyrskjot	E1 = K3	0,55	0,55	0,55	0,55	0,5	0,55	0,54	0,54	0,54
eTongueSugar	E1 = K2	0,7	0,68	0,68	0,68	0,68	0,7	0,64	0,68	▽**0,54**
	E2 = K3	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,03
golub	E1 = K2	0,94	0,94	0,94	0,94	0,88	0,94	0,94	0,88	▽**0,61**
	E2 = K2	0,31	▽**0,12**	0,31	0,31	0,31	0,31	0,31	0,31	0,31
	E3 = K3	0,87	0,87	0,87	▽**0,68**	0,8	▽**0,68**	0,8	0,87	0,87
	E4 = K4	0,73	0,73	0,73	0,73	0,67	0,73	0,73	0,69	▽**0,46**
gordon	E1 = K2	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17
leukemia	E1 = K3	0,33	0,33	0,33	0,33	0,29	0,33	0,29	0,33	0,27
	E2 = K7	0,77	▽**0,22**	0,76	0,77	0,75	0,71	0,77	0,77	0,75
lung	E1 = K4	0,64	0,64	0,54	▽**0,43**	0,64	▽**0,41**	0,64	0,54	▽**0,52**
miRNAcancer	E1 = K2	0,59	0,59	0,59	0,59	0,57	0,59	0,57	0,54	▽**0,47**
	E2 = K2	0,22	0,22	0,14	0,14	0,22	0,22	▽**0,10**	0,13	▽**0,09**
	E3 = K3	0,3	0,3	0,3	0,3	0,3	0,3	0,24	0,2	0,23
	E4 = K4	0,66	0,66	0,66	0,66	0,66	0,66	0,64	0,66	0,57
	E5 = K9	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,25
	E6 = K20	0,42	0,42	0,42	0,42	0,4	0,4	0,42	0,42	0,4
Média		0,72	0,64	0,68	0,66	0,70	0,71	0,69	0,69	0,66
Perda média de qualidade em relação a Π_{Rep}			11,81%	6,31%	8,62%	3,18%	2,27%	4,01%	5,13%	8,77%

Tabela 5 – Médias e desvios padrão do número de partições resultantes das múltiplas execuções do MOCLE sob a influência das diferentes seleções.

Dataset	$\lvert Pi_{Rep}\rvert$	*Baseline*	SRD	SR	BRP	ASA	Diversidade	CAS	FILTA	HSS
atom	21	12,1 ± 0,9	**▽6,6 ± 0,6**	**▽6,5 ± 0,6**	**▽6,3 ± 0,5**	**▽6,8 ± 0,9**	**▽6,3 ± 0,4**	**▽9,7 ± 0,6**	**▽8,0 ± 0,8**	**▽6,0 ± 0,8**
2sp2glob	54	18,9 ± 1,2	**▽7,3 ± 0,5**	**▽7,3 ± 0,4**	**▽7,3 ± 0,5**	**▽9,2 ± 0,8**	**▽13,0 ± 1,3**	**▽9,0 ± 0,0**	**18,8 ± 1,0**	**▽12,7 ± 0,7**
aggregation	74	33,3 ± 4,2	**30,9 ± 3,1**	**▽27,8 ± 2,5**	**▽19,4 ± 2,0**	**▽16,5 ± 2,6**	**▽25,6 ± 2,7**	**▽21,5 ± 2,6**	**△36,6 ± 4,3**	**▽11,3 ± 1,7**
chainlink	27	9,0 ± 0,2	**▽8,0 ± 0,9**	**▽6,2 ± 0,6**	**▽8,0 ± 0,7**	**▽6,1 ± 0,2**	8,1 ± 1,2	**▽6,1 ± 0,3**	8,1 ± 0,2	**▽6,0 ± 0,0**
compound	62	23,2 ± 2,7	**▽11,6 ± 1,0**	**▽11,6 ± 1,0**	**▽11,7 ± 0,9**	**▽15,4 ± 1,6**	**▽16,4 ± 3,2**	**▽15,0 ± 1,4**	**23,7 ± 3,6**	**▽12,2 ± 1,5**
ds2c2sc13	147	67,3 ± 6,6	**▽44,0 ± 3,7**	**▽45,6 ± 4,0**	**▽45,9 ± 3,6**	**▽18,8 ± 3,6**	**▽35,2 ± 5,7**	**▽56,1 ± 6,1**	**▽19,6 ± 1,1**	**▽25,1 ± 2,6**
flame	25	14,7 ± 1,5	**▽12,1 ± 1,4**	**▽11,8 ± 1,5**	**▽12,0 ± 0,9**	**15,7 ± 1,7**	**▽11,3 ± 1,7**	**▽9,8 ± 0,7**	**▽12,3 ± 1,5**	**▽7,7 ± 0,6**
gaussian3	45	4,0 ± 0,0	**4,0 ± 0,0**	**4,0 ± 0,0**	**▽3,0 ± 0,0**	**△5,0 ± 0,0**	3,3 ± 0,7	**4,0 ± 0,0**	**4,0 ± 0,0**	**4,0 ± 0,0**
hepta	88	17,5 ± 1,1	**▽9,0 ± 0,2**	**▽12,1 ± 0,2**	**▽9,1 ± 0,2**	**▽7,7 ± 1,2**	**▽11,9 ± 1,3**	**▽11,6 ± 0,7**	**▽11,8 ± 0,8**	**▽12,8 ± 0,6**
jain	27	5,6 ± 0,7	**5,7 ± 0,7**	6,3 ± 0,9	**5,5 ± 1,0**	**5,6 ± 0,6**	**▽4,4 ± 0,6**	**▽4,3 ± 0,5**	**5,4 ± 0,7**	**▽4,2 ± 0,6**
lsun	29	13,0 ± 1,4	**▽11,8 ± 2,5**	**▽11,6 ± 1,9**	**11,9 ± 1,8**	**▽5,7 ± 0,7**	**▽7,3 ± 0,9**	**▽10,2 ± 1,5**	**12,2 ± 1,2**	**▽6,3 ± 0,6**
monkey	76	31,2 ± 7,3	**▽24,2 ± 3,8**	**▽26,1 ± 4,9**	**▽18,5 ± 4,6**	**▽15,7 ± 2,8**	**▽19,6 ± 2,9**	**▽24,7 ± 4,6**	**31,6 ± 6,6**	**▽8,2 ± 1,5**
pathbased	29	12,2 ± 1,6	**▽7,2 ± 0,8**	**▽7,0 ± 0,0**	**▽8,1 ± 0,2**	**▽9,1 ± 0,9**	**▽9,4 ± 1,3**	**▽8,1 ± 1,9**	**▽9,3 ± 2,0**	**▽5,3 ± 0,5**
R15	154	56,3 ± 3,8	**▽34,8 ± 3,1**	**▽37,6 ± 3,5**	**▽37,0 ± 2,6**	**▽8,4 ± 1,0**	**▽38,1 ± 4,9**	**▽41,3 ± 3,5**	**▽34,2 ± 3,5**	**▽33,6 ± 2,8**
simulated6	58	11,0 ± 0,0	**▽7,0 ± 0,2**	**▽8,0 ± 0,0**	**▽7,0 ± 0,0**	**▽3,9 ± 0,4**	**▽4,8 ± 0,7**	**11,0 ± 0,2**	**11,1 ± 0,2**	**▽6,6 ± 0,5**
spiralsquare	85	24,1 ± 1,6	**▽8,1 ± 0,3**	**▽8,1 ± 0,2**	**▽9,3 ± 0,6**	**▽9,6 ± 2,0**	**▽21,6 ± 3,2**	**▽9,2 ± 0,4**	**25,8 ± 3,0**	**▽21,5 ± 1,8**
target	55	34,8 ± 5,8	**▽12,6 ± 2,5**	**▽13,7 ± 1,6**	**▽21,9 ± 3,5**	**▽21,4 ± 4,3**	**▽18,6 ± 3,6**	**▽26,1 ± 3,4**	**▽27,7 ± 3,9**	**▽14,7 ± 2,6**
tetra	53	22,7 ± 1,1	**▽13,1 ± 0,2**	**▽14,3 ± 0,5**	**▽14,4 ± 0,5**	**▽5,0 ± 0,9**	**▽10,7 ± 1,7**	**▽17,5 ± 0,7**	**▽19,3 ± 1,0**	**▽10,0 ± 1,0**
twoDiamonds	24	9,1 ± 0,3	**▽2,0 ± 0,0**	**▽6,0 ± 0,5**	**▽5,0 ± 0,0**	**▽5,0 ± 0,0**	**▽5,4 ± 0,6**	**▽6,9 ± 0,2**	**▽5,9 ± 0,7**	**▽7,0 ± 0,8**
wingNut	17	16,7 ± 2,6	**▽13,0 ± 1,4**	**▽14,0 ± 1,5**	**▽11,9 ± 1,7**	**▽7,5 ± 1,3**	**▽11,5 ± 1,7**	**▽10,7 ± 0,9**	**▽14,2 ± 2,1**	**▽5,6 ± 0,8**
glass	55	29,2 ± 3,3	**▽19,2 ± 1,5**	**▽19,3 ± 1,8**	**▽20,1 ± 2,9**	**▽11,7 ± 2,1**	**▽16,4 ± 2,8**	**▽23,9 ± 3,9**	**▽14,2 ± 2,1**	**▽16,6 ± 2,3**
iris	37	9,5 ± 0,6	**▽3,0 ± 0,0**	**▽3,0 ± 0,0**	**▽5,0 ± 0,0**	**▽5,5 ± 0,6**	**9,3 ± 1,3**	**▽3,0 ± 0,0**	10,1 ± 0,7	**▽7,1 ± 0,8**
chowdary	15	5,0 ± 0,0	**▽4,0 ± 0,0**	**▽3,6 ± 0,5**	**▽4,0 ± 0,0**	**▽4,0 ± 0,0**	**5,0 ± 0,0**	**▽4,0 ± 0,0**	**▽4,0 ± 0,0**	**▽3,1 ± 0,2**
dyrskjot	25	7,3 ± 0,4	**7,5 ± 0,7**	**▽5,0 ± 0,2**	**▽6,1 ± 0,4**	**▽4,9 ± 0,6**	6,5 ± 1,0	**7,0 ± 0,2**	**7,6 ± 0,8**	6,4 ± 0,6
eTongueSugar	25	7,2 ± 1,7	**▽5,6 ± 1,1**	**7,4 ± 1,5**	**7,8 ± 1,3**	**▽5,0 ± 0,9**	**▽5,8 ± 0,9**	**7,2 ± 0,9**	**7,4 ± 1,7**	**▽3,7 ± 1,0**
golub	49	14,8 ± 1,0	**▽11,8 ± 1,9**	**▽10,2 ± 0,6**	**▽13,1 ± 1,4**	**▽10,5 ± 1,0**	**14,0 ± 2,1**	**▽12,9 ± 1,0**	**15,0 ± 0,7**	**▽10,2 ± 1,0**
gordon	15	11,0 ± 1,2	**▽8,9 ± 0,8**	**▽7,3 ± 0,8**	**▽7,4 ± 0,7**	**▽8,4 ± 0,6**	**▽7,5 ± 0,5**	**▽7,2 ± 0,7**	**▽9,6 ± 1,0**	**▽4,6 ± 0,6**
leukemia	66	18,6 ± 1,7	**▽6,6 ± 0,5**	**▽6,7 ± 0,4**	**▽9,6 ± 0,5**	**▽10,0 ± 1,9**	**▽13,6 ± 1,8**	**▽15,5 ± 1,7**	**19,1 ± 1,7**	**▽9,0 ± 0,9**
lung	40	18,2 ± 3,3	**▽7,1 ± 0,5**	**▽10,2 ± 0,8**	**▽9,3 ± 0,5**	**▽10,9 ± 2,8**	**▽12,6 ± 2,6**	**▽14,3 ± 2,8**	**17,9 ± 3,2**	**▽8,7 ± 1,7**
miRNAcancer	218	78,6 ± 5,6	**▽73,8 ± 5,4**	**78,6 ± 5,3**	**79,8 ± 4,6**	**▽27,2 ± 3,8**	**▽36,2 ± 4,8**	**▽73,5 ± 6,2**	**81,7 ± 8,1**	**▽26,4 ± 4,4**
Média	56,50	21,20	14,02	14,56	14,51	9,87	13,65	16,04	17,54	10,55
Redução em relação ao *Baseline*			33,89%	31,32%	31,55%	53,43%	35,64%	24,34%	17,28%	50,23%

Antes de discutir e comparar os resultados de cada um dos métodos de seleção, uma visão geral do impacto das seleções na qualidade dos resultados do MOCLE, pode ser obtida através de um resumo da Tabela 3, contido na Tabela 6. Nela está uma contagem das diferentes variações de desempenho observadas.

Tabela 6 – Resumo da Tabela 3.

Tipo	Número de casos	Porcentagem do Total
Estatisticamente similar	272	70.83%
Melhora relevante	1	0.26%
Piora relevante	31	8.07%
Variação não relevante	80	20.84%
Total	384	100%

Na Tabela 6 é possível observar que não houve forte influência das seleções na capacidade do MOCLE em encontrar partições de alta qualidade (previamente conhecidas). Pois, em 91,67% dos casos contidos na Tabela 3 não foi observada variação relevante ou os eles foram estatisticamente similares ao *baseline*. Além disso, houve apenas um caso de melhora relevante e os casos de piora representam apenas 8,07% do total.

5.1.1 *Baseline*

A Tabela 3 revela que mesmo o *framework* usando todo o repositório (que é o *baseline*), o MOCLE encontra dificuldade em melhorar as partições que lhe foram fornecidas, isto é, a qualidade não aumenta em relação as melhores partições do repositório (Π_{Rep}) — pelo menos, para os *datasets* testados, sob as condições descritas no Capítulo 4. Para as 48 partições conhecidas dos 30 *datasets* analisados, apenas em uma houve melhora maior ou igual a 0,1 na média (no `lung`, sendo a melhor partição de $ARI = 0{,}826$). Porém, em outros dois *datasets* houve melhorias em identificar as partições conhecidas em algumas das 30 execuções no *baseline*, no `gordon` a melhor partição encontrada foi de $ARI = 0{,}633$ e para o `eTongueSugar` a melhor foi $ARI = 0{,}895$. Para mais, em seis casos foram observadas pioras relevantes, isto é, com perda maior ou igual a $-0{,}1$. Esses casos de piora foram apenas uma fração dos casos analisados e observando a perda média é possível notar que, embora presente, a perda não foi significativa. Além disso, é importante notar que houve uma redução de 62,47% no número de partições em relação ao Π_{Rep}, na Tabela 5.

Em suma, para os *datasets* testados, o MOCLE, sem influência das seleções, reduziu o número de partições sem grandes impactos — positivos ou negativos — na qualidade dessas partições.

5.1.2 SR, SRD, BRP e Diversidade

A Tabela 3 revela que, de forma geral, nenhum método de seleção teve um impacto fortemente negativo na qualidade dos resultados do MOCLE, a ponto de inviabilizar sua utilização. Porém, os métodos SR, SRD, BRP e Diversidade resultaram em perdas de qualidade relevantes em relação ao *baseline*. Essas quatro seleções são responsáveis por 80,64% de todos os casos de piora identificados — aqueles destacados em vermelho. Dentre elas, o SRD foi a seleção que mais impactou negativamente, com 9,31% de perda média, já o SR causou o menor impacto com 4,30% de perda média.

O que destaca essas seleções é que, em diversos casos, elas pioraram resultados de alta qualidade (média do ARI >= 0,8) presentes no *baseline*. Exemplos disso, são os casos dos *datasets*: `atom`, `chainlink`, `jain` e `golub`. Uma explicação para essa piora é que esses métodos não conseguiram fornecer partições relevantes ao MOCLE. E isso, por consequência, gerou um conjunto de baixa qualidade. Outra explicação possível é que, mesmo fornecendo boas partições, o MOCLE não conseguiu identificá-las nesses conjuntos reduzidos. Na primeira explicação possível, a causa da perda de desempenho é responsabilidade dos métodos de seleção e na segunda uma deficiência do MOCLE.

Para explorar essas duas possibilidades, foi investigado, assim como feito em Faceli (2006), o conjunto de soluções geradas pelo MOCLE, por meio da análise da origem dessas partições. Essa análise foi feita por gráficos que mostram a variância e a conectividade das partições — que são as funções objetivo do MOCLE —, e também a fronteira de Pareto obtida com essas partições[1]. Para a construção desses gráficos, os valores foram normalizados de forma igual a Faceli (2006), seguindo as observações de Handl e Knowles (2007). Isto é, primeiro todos os valores de conectividade e variância foram normalizados no intervalo $[0, 1]$ e depois foi extraída a raiz quadrada de cada valor. Segundo Handl e Knowles (2007), a radiciação é desejável para mitigar distorções causadas na plotagem dos valores. Tais distorções são causadas devido ao comportamento não linear da conectividade e da variância em relação ao número de *clusters* das partições.

Todos os casos de piora relevante foram investigados seguindo essa abordagem. As Figuras 6, 7 e 8 exemplificam o que aconteceu nos *datasets* `golub`, `jain` e `atom`, respectivamente, e representam o que aconteceu em todos os outros.

A Figura 6 mostra como ocorreu a perda de qualidade para a seleção Diversidade na estrutura E1 do `golub`. A Figura 6a mostra que a partição com maior qualidade (ARI = 0,94), em relação a E1, ficou fora da fronteira de Pareto e por isso não foi considerada pelo MOCLE. Porém, uma outra partição de qualidade similar (ARI = 0,88), que ficou na fronteira, foi preservada durante o processo de evolução e incluída em todos os 30 conjuntos de solução. Por outro lado, na Figura 6b é possível ver que mesmo que a Diversidade tenha

[1] As partições fornecidas ao MOCLE são selecionadas através da fronteira de Pareto, as que se encontram fora da fronteira não são consideradas na saída ou em combinações do processo de otimização.

selecionado a partição de maior qualidade disponível (ARI = 0,94), a mesma também ficou fora da fronteira, deixando apenas partições de pouca qualidade nela (ARI <= 0,24). Isso demonstra que, nesse caso, o MOCLE não incluiu uma partição de alta qualidade em seu conjunto de resultados, mesmo recebendo menos partições.

A Figura 7 também mostra outro caso de perda de qualidade, utilizando o BRP para o *dataset* `jain`. Nesse caso, foi o BRP que não identificou a partição de maior qualidade (ARI = 1), pois na Figura 7b consta que a melhor partição selecionada teve ARI = 0,71. Contudo, para piorar, o MOCLE também falhou em identificar a melhor partição disponível, pois esta não estava na fronteira de Pareto; a que estava, e foi utilizada, tinha qualidade inferior (ARI = 0,51).

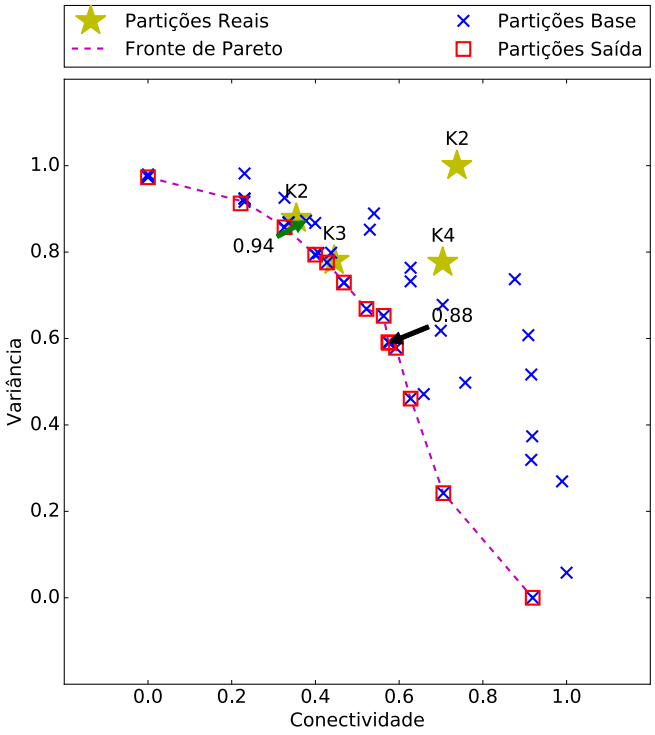
Já a Figura 8 exemplifica o que ocorreu na maioria dos casos de piora[2] — incluindo ou não os casos de piora do HSS e do ASA. A Figura 8b demonstra como a seleção, nesse caso o SRD, ignorou uma ou mais partições de alta qualidade e o MOCLE, por sua vez, conseguiu identificar a melhor selecionada e mantê-la no conjunto de soluções ou fazer outras combinações utilizando-a.

A Tabela 4 corrobora essas observações sobre a responsabilidade da piora ser, em grande parte, dos métodos de seleção. Nela é possível ver que há falhas em identificar as melhores partições disponíveis em diversos casos — principalmente para o SRD, SR e BRP — e que isso reflete negativamente na performance do MOCLE na Tabela 3. Em uma análise mais profunda, o que explica esse impacto negativo causado pelas seleções SR, SRD e BRP é uma característica transitória dos métodos, que são os índices relativos considerados pelos mesmos. No presente trabalho, para evitar possíveis vieses causados pela alteração dos mesmos, optou-se por manter os seis índices adotados em Naldi, Carvalho e Campello (2013). Porém, todos esses índices dão preferência para partições com *clusters* bem separados e compactos. O que leva a esses três métodos de seleção a ter um forte viés para partições compostas por *clusters* com essas características. Por isso, em alguns dos *datasets* aqui considerados houve perda de performance, já que não são todos que possuem estruturas reais com essas características.

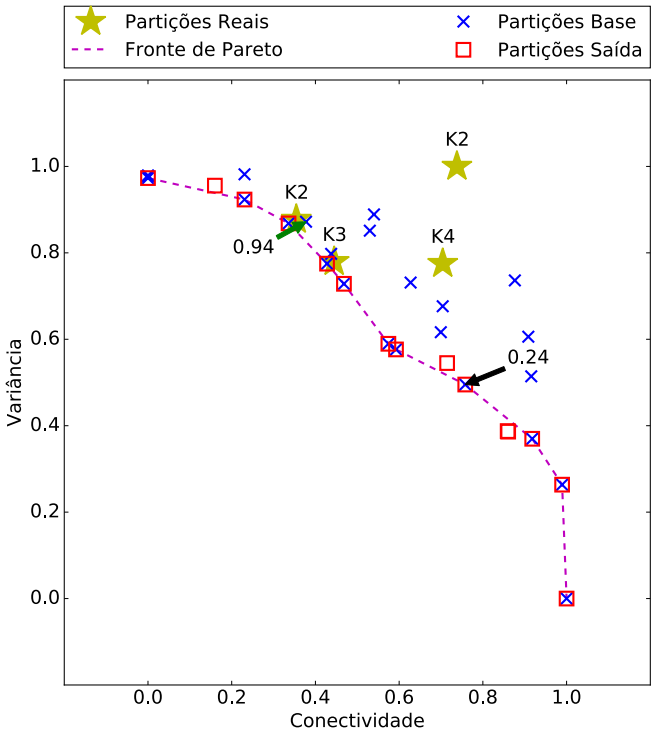
Diferentemente das seleções SR, SRD e BRP, a Tabela 4 indica que a Diversidade foi a melhor seleção em identificar partições de alta qualidade em relação as estruturas conhecidas; superando, inclusive, as seleções que tiveram melhor impacto em termos de qualidade no MOCLE (ASA, CAS e FILTA). Essa observação aliada a análise da fronteira de Pareto indica que o MOCLE tende a gerar resultados de pior qualidade quando a composição de seu conjunto de partições de entrada (Π_I) é pautada apenas pela diversidade entre seus membros — assim como *ensembles* e outros trabalhos de *multiple clustering* (HADJITODOROV; KUNCHEVA; TODOROVA, 2006; FERN; LIN, 2008; MULLER et al., 2010; NALDI; CARVALHO; CAMPELLO, 2013).

[2] Aqueles destacados em vermelho na Tabela 3

Figura 6 – Fronteiras de Pareto para o *dataset* `golub` (com e sem seleção). A seta verde destaca a partição fornecida ao MOCLE mais parecida com E1. A seta preta destaca a partição mais parecida com E1 no Fronte de Pareto. Ambas setas informam o ARI delas em relação a E1.



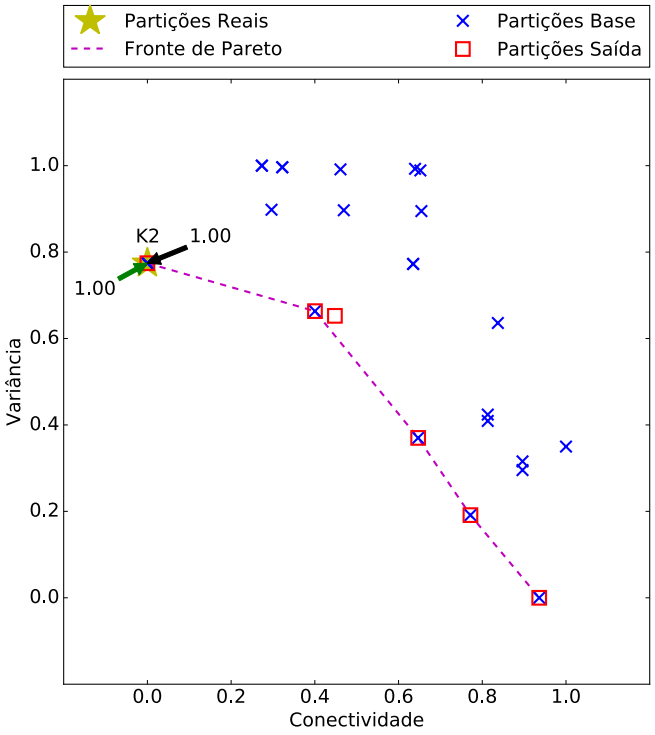
(a) *Baseline*, as partições base são as de Π_{Rep}.



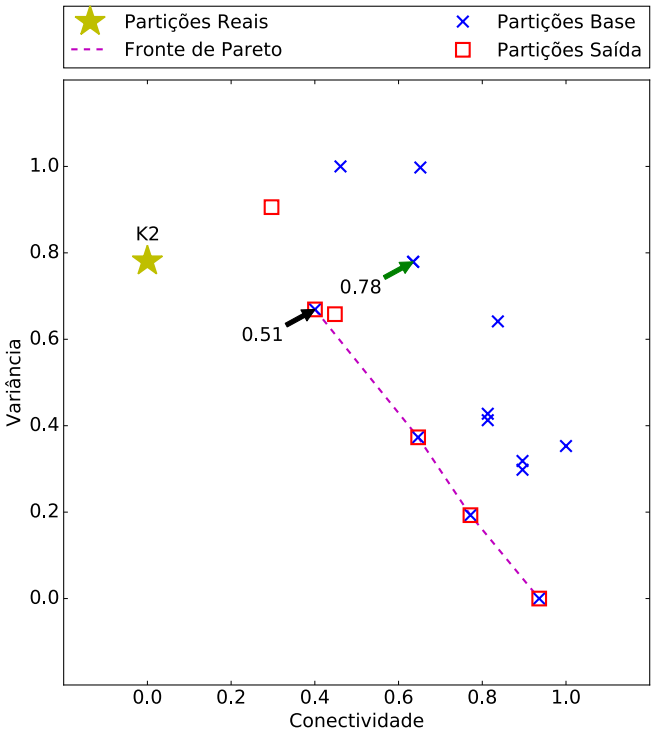
(b) Diversidade, as partições base são as de Π_R.

Fonte: elaboração própria.

Figura 7 – Fronteiras de Pareto para o *dataset* `jain` (com e sem seleção). A seta verde destaca a partição fornecida ao MOCLE mais parecida com E1. A seta preta destaca a partição mais parecida com E1 no Fronte de Pareto. Ambas setas informam o ARI delas em relação a E1.



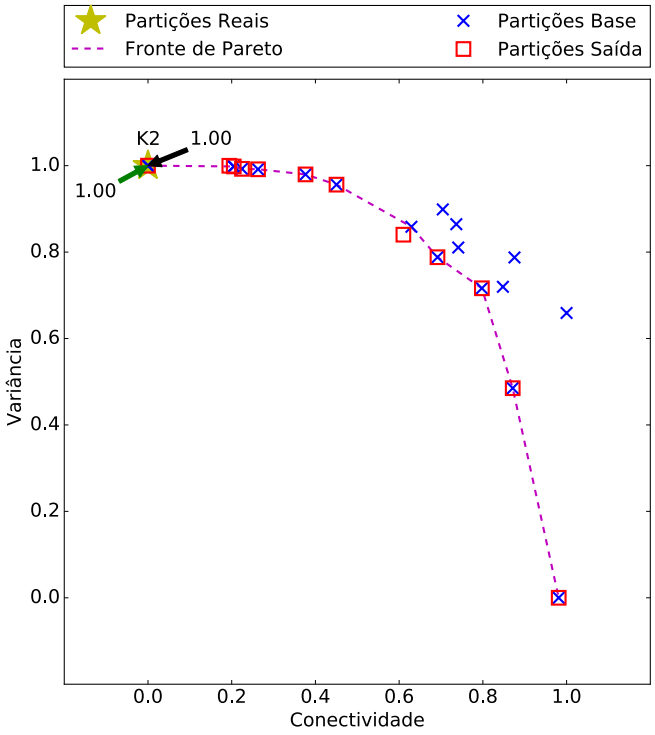
(a) *Baseline*, as partições base são as de Π_{Rep}.



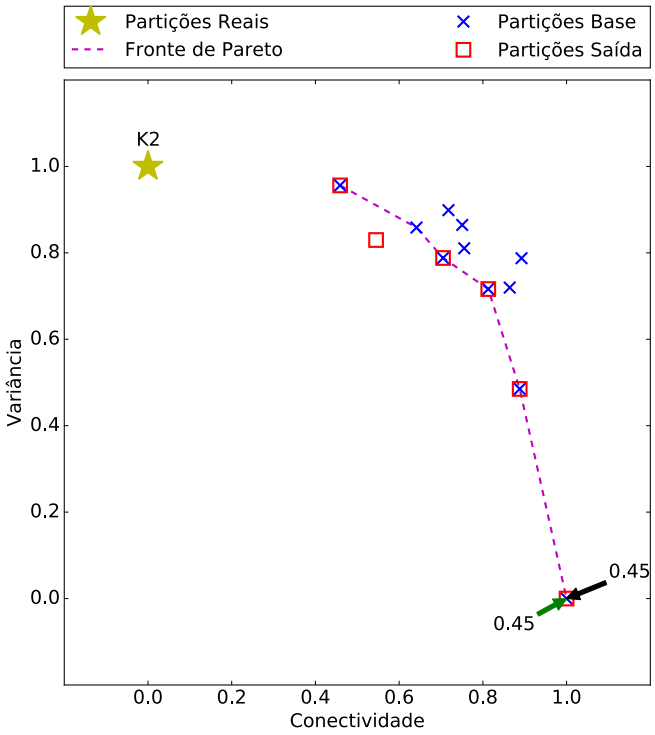
(b) BRP, as partições base são as de Π_R.

Fonte: elaboração própria.

Figura 8 – Fronteiras de Pareto para o *dataset* `atom` (com e sem seleção). A seta verde destaca a partição fornecida ao MOCLE mais parecida com E1. A seta preta destaca a partição mais parecida com E1 no Fronte de Pareto. Ambas setas informam o ARI delas em relação a E1.



(a) *Baseline*, as partições base são as de Π_{Rep}.



(b) SRD, as partições base são as de Π_R.

Fonte: elaboração própria.

5.1.3 HSS

A Tabela 3 revela que o HSS causou um impacto negativo no MOCLE, especialmente em *datasets* reais. No entanto, diferentemente do SR, BRP, SRD e Diversidade, exceto por um caso (`golub`, na E1), essa perda de qualidade não foi em casos em que o ARI ultrapassa 0,8 no *baseline*. Além disso, o HSS obteve esses resultados fornecendo ao MOCLE menos da metade das partições que essas outras quatro seleções forneceram (em média), assim como a Tabela 7 indica. Essa redução na entrada do número de partições do MOCLE refletiu em uma saída mais enxuta — cerca de 50% menor que o *baseline*. Esses resultados indicam que o HSS conseguiu auxiliar o MOCLE a encontrar um conjunto de resultados conciso sem perdas relevantes de qualidade. Superando os resultados obtidos com as seleções SR, BRP, SRD e Diversidade.

Vale lembrar que esses resultados foram obtidos utilizando os parâmetros padrão dessa seleção. Ajustes, em que o número de partições selecionadas seja maior, talvez sejam suficientes para amenizar as perdas de qualidade e manter uma redução significativa no número de partições finais do MOCLE.

Tabela 7 – Número de partições selecionadas pelos diferentes métodos.

Dataset	$\lvert\Pi_{Rep}\rvert$	SRD, SR, BRP, Diversidade, CAS e FILTA	ASA	HSS
`atom`	21	11	10	4
`2sp2glob`	54	27	12	13
`aggregation`	74	37	14	10
`chainlink`	27	14	14	5
`compound`	62	31	13	8
`ds2c2sc13`	147	74	18	18
`flame`	25	13	18	5
`gaussian3`	45	23	13	6
`hepta`	88	44	30	15
`jain`	27	14	16	3
`lsun`	29	15	9	5
`monkey`	76	38	24	7
`pathbased`	29	15	11	5
`R15`	154	77	25	23
`simulated6`	58	29	12	7
`spiralsquare`	85	43	9	15
`target`	55	28	24	7
`tetra`	53	27	9	8
`twoDiamonds`	24	12	11	4
`wingNut`	17	9	7	3
`glass`	55	28	9	9
`iris`	37	19	7	6
`chowdary`	15	8	4	3
`dyrskjot`	25	13	7	7
`eTongueSugar`	25	13	10	3
`golub`	49	25	25	9
`gordon`	15	8	9	4
`leukemia`	66	33	23	6
`lung`	40	20	17	5
`miRNAcancer`	218	109	56	19
Média	56,50	21,20	15,53	8,07
Redução em relação ao Π_{Rep}		49,44%	72,51%	85,72%

5.1.4 ASA, CAS e FILTA

O ASA, CAS e o FILTA foram as três melhores seleções em termos de qualidade na Tabela 3. Todas as três quase se igualaram ao *baseline*. Dentre elas, houve apenas uma perda e um ganho relevante de qualidade (nos *datasets* `dyrskjot` e `gordon` respectivamente), ambos com partições selecionadas pelo ASA[3]. Nos dois casos, em maior análise se constatou que o ASA selecionou as melhores partições possíveis e a alteração de performance foi causada pelo MOCLE que as recombinou com os outros agrupamentos disponíveis.

No entanto, a performance similar dos três métodos não se mantém quando são analisados os números de partições geradas pelo MOCLE. Os resultados contidos na Tabela 5 indicam que o ASA levou o MOCLE a, em média, gerar quase metade das partições do CAS e do FILTA. Porém, é injusto comparar essas três seleções nesses termos, pois o ASA forneceu significantemente menos partições ao MOCLE, como consta na Tabela 7. Por isso, uma comparação adicional entre os três métodos foi feita, na qual, as seleções CAS e FILTA foram parametrizados de forma a selecionar o mesmo número de partições que o ASA para cada *dataset*. O resultado dessa comparação se encontram nas Tabelas 8 e 9, onde a mesma categorização e padrão de cores das Tabelas 3 e 5 foi respectivamente adotado.

A Tabela 8 revela que para o FILTA, a redução no número de partições selecionadas piorou os resultados da qualidade, principalmente no caso da E2 para o *dataset* `ds2c2sc13`. Porém, esses resultados foram melhores do que aqueles obtidos pelo SR, SRD, BRP e Diversidade que utilizaram mais partições. Isso revela que o FILTA tolera reduções sem grandes prejuízos e mantém sua habilidade de selecionar boas partições. Já para o CAS, o resultado foi ainda melhor, ultrapassando em algumas frações o resultado do ASA e se aproximando ainda mais do *baseline*.

Considerando a redução no número de partições selecionadas, o FILTA causou apenas uma leve melhora no número de partições geradas pelo MOCLE, como consta na Tabela 9. O CAS, por sua vez, quase igualou o número de partições geradas com o uso do ASA.

[3] A melhoria no *dataset* `gordon` é a única identificada em todos os casos, seja para o *baseline* ou para as seleções.

Tabela 8 – Médias e desvios padrão do ARI das melhores partições resultantes das múltiplas execuções do MOCLE sob a influência das diferentes seleções. Análise das melhores: ASA, CAS e FILTA.

Dataset	Estrutura	Π_{Rep}	*Baseline*	ASA	CAS	FILTA
atom	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
2sp2glob	E1 = K4	1	0.71 ± 0.00	**0.71 ± 0.00**	**0.71 ± 0.00**	**0.71 ± 0.00**
aggregation	E1 = K7	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	0.94 ± 0.05
chainlink	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
compound	E1 = K6	0.85	0.81 ± 0.03	**0.81 ± 0.03**	**0.80 ± 0.01**	**0.79 ± 0.03**
ds2c2sc13	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	E2 = K5	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**▽0.62 ± 0.00**
	E3 = K13	1	0.68 ± 0.04	0.76 ± 0.00	**△0.78 ± 0.02**	0.59 ± 0.00
flame	E1 = K2	0.94	0.97 ± 0.01	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.97 ± 0.01**
gaussian3	E1 = K3	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
hepta	E1 = K7	0.97	0.97 ± 0.00	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.97 ± 0.00**
jain	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
lsun	E1 = K3	1	1.00 ± 0.00	**0.98 ± 0.05**	**0.99 ± 0.05**	**1.00 ± 0.00**
monkey	E1 = K2	0.73	0.55 ± 0.05	0.60 ± 0.11	**0.57 ± 0.07**	**0.53 ± 0.09**
	E2 = K3	0.83	0.69 ± 0.04	**0.68 ± 0.02**	**0.68 ± 0.03**	**0.68 ± 0.04**
	E3 = K5	0.65	0.57 ± 0.02	**0.59 ± 0.02**	**0.59 ± 0.03**	**0.59 ± 0.04**
	E4 = K8	0.67	0.66 ± 0.02	0.65 ± 0.00	**0.67 ± 0.00**	**0.66 ± 0.02**
pathbased	E1 = K3	0.49	0.49 ± 0.00	0.47 ± 0.01	**0.49 ± 0.00**	**0.49 ± 0.01**
R15	E1 = K15	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	**▽0.87 ± 0.00**
simulated6	E1 = K6	0.99	0.99 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	0.96 ± 0.01
spiralsquare	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.98 ± 0.07**
	E2 = K6	0.67	0.67 ± 0.00	**0.67 ± 0.00**	0.66 ± 0.00	0.63 ± 0.03
target	E1 = K6	1	0.97 ± 0.00	**0.93 ± 0.10**	**0.97 ± 0.00**	**0.97 ± 0.01**
tetra	E1 = K4	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
twoDiamonds	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
wingNut	E1 = K2	1	1.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
glass	E1 = K2	0.67	0.67 ± 0.01	**0.67 ± 0.00**	**0.67 ± 0.00**	**▽0.25 ± 0.04**
	E2 = K5	0.56	0.56 ± 0.01	0.55 ± 0.00	0.55 ± 0.00	**▽0.23 ± 0.02**
	E3 = K6	0.26	0.26 ± 0.01	**0.26 ± 0.00**	**0.26 ± 0.00**	0.19 ± 0.00
iris	E1 = K3	0.82	0.76 ± 0.00	**0.76 ± 0.00**	**0.76 ± 0.00**	0.76 ± 0.00
chowdary	E1 = K2	0.07	0.07 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00
dyrskjot	E1 = K3	0.55	0.54 ± 0.00	**▽0.38 ± 0.00**	**▽0.38 ± 0.00**	0.49 ± 0.00
eTongueSugar	E1 = K2	0.7	0.72 ± 0.11	**0.62 ± 0.05**	**0.66 ± 0.08**	**0.71 ± 0.12**
	E2 = K3	0.05	0.07 ± 0.01	0.05 ± 0.01	**0.07 ± 0.01**	**0.07 ± 0.01**
golub	E1 = K2	0.94	0.88 ± 0.00	**0.88 ± 0.00**	**0.88 ± 0.00**	**0.88 ± 0.00**
	E2 = K2	0.31	0.31 ± 0.00	**0.31 ± 0.00**	**0.31 ± 0.00**	**0.31 ± 0.00**
	E3 = K3	0.87	0.87 ± 0.01	0.80 ± 0.00	0.80 ± 0.01	**0.88 ± 0.01**
	E4 = K4	0.73	0.67 ± 0.00	**0.67 ± 0.00**	**0.67 ± 0.01**	**0.67 ± 0.00**
gordon	E1 = K2	0.17	0.22 ± 0.13	**△0.39 ± 0.13**	**0.22 ± 0.11**	**0.19 ± 0.07**
leukemia	E1 = K3	0.33	0.41 ± 0.00	**0.39 ± 0.05**	**▽0.30 ± 0.00**	0.33 ± 0.05
	E2 = K7	0.77	0.77 ± 0.00	0.75 ± 0.00	**0.77 ± 0.00**	**0.77 ± 0.00**
lung	E1 = K4	0.64	0.77 ± 0.06	**0.73 ± 0.09**	0.71 ± 0.08	**0.71 ± 0.10**
miRNAcancer	E1 = K2	0.59	0.62 ± 0.01	**0.63 ± 0.05**	**0.62 ± 0.03**	**0.63 ± 0.05**
	E2 = K2	0.22	0.10 ± 0.01	**0.11 ± 0.03**	0.09 ± 0.01	0.09 ± 0.03
	E3 = K3	0.3	0.23 ± 0.01	0.22 ± 0.01	**0.23 ± 0.01**	0.21 ± 0.02
	E4 = K4	0.66	0.61 ± 0.03	**0.61 ± 0.04**	**0.60 ± 0.03**	**0.63 ± 0.05**
	E5 = K9	0.3	0.32 ± 0.05	0.28 ± 0.04	**0.30 ± 0.05**	0.30 ± 0.06
	E6 = K20	0.42	0.42 ± 0.00	0.40 ± 0.00	0.41 ± 0.00	0.41 ± 0.00
Média		0.72	0.70	0.69	0.69	0.66
Perda média de qualidade em relação ao *Baseline*				0.84%	0.46%	4.23%

Tabela 9 – Médias e desvios padrão do número de partições resultantes das múltiplas execuções do MOCLE sob a influência das diferentes seleções. Análise dos melhores: ASA, CAS e FILTA.

Dataset	$\lvert\Pi_{Rep}\rvert$	*Baseline*	ASA	CAS	FILTA
atom	21	12,1 ± 0,9	**▽6,8 ± 0,9**	**▽9,9 ± 0,9**	**▽6,5 ± 0,8**
2sp2glob	54	18,9 ± 1,2	**▽9,2 ± 0,8**	**▽9,0 ± 0,9**	**▽11,2 ± 1,2**
aggregation	74	33,3 ± 4,2	**▽16,5 ± 2,6**	**▽17,8 ± 2,1**	**▽20,9 ± 2,4**
chainlink	27	9,0 ± 0,2	**▽6,1 ± 0,2**	**▽6,1 ± 0,2**	8,1 ± 0,3
compound	62	23,2 ± 2,7	**▽15,4 ± 1,6**	**▽9,7 ± 1,1**	**▽16,1 ± 2,4**
ds2c2sc13	147	67,3 ± 6,6	**▽18,8 ± 3,6**	**▽20,4 ± 2,2**	**▽3,0 ± 0,0**
flame	25	14,7 ± 1,5	**15,7 ± 1,7**	**14,0 ± 1,3**	**14,4 ± 1,8**
gaussian3	45	4,0 ± 0,0	**△5,0 ± 0,0**	**4,0 ± 0,0**	**3,8 ± 0,4**
hepta	88	17,5 ± 1,1	**▽7,7 ± 1,2**	**▽11,4 ± 1,2**	**▽10,3 ± 0,6**
jain	27	5,6 ± 0,7	**5,6 ± 0,6**	**▽4,2 ± 0,5**	**5,7 ± 0,9**
lsun	29	13,0 ± 1,4	**▽5,7 ± 0,7**	**▽6,9 ± 1,2**	**▽7,1 ± 1,2**
monkey	76	31,2 ± 7,3	**▽15,7 ± 2,8**	**▽14,7 ± 3,1**	**27,1 ± 4,9**
pathbased	29	12,2 ± 1,6	**▽9,1 ± 0,9**	**▽6,1 ± 1,2**	**▽8,1 ± 1,4**
R15	154	56,3 ± 3,8	**▽8,4 ± 1,0**	**▽16,5 ± 2,2**	**▽14,8 ± 1,9**
simulated6	58	11,0 ± 0,0	**▽3,9 ± 0,4**	**▽5,2 ± 0,4**	**▽6,7 ± 0,6**
spiralsquare	85	24,1 ± 1,6	**▽9,6 ± 2,0**	**▽7,4 ± 1,2**	**▽14,6 ± 2,7**
target	55	34,8 ± 5,8	**▽21,4 ± 4,3**	**▽21,5 ± 3,4**	**▽24,3 ± 3,4**
tetra	53	22,7 ± 1,1	**▽5,0 ± 0,9**	**▽6,6 ± 0,5**	**▽12,3 ± 1,1**
twoDiamonds	24	9,1 ± 0,3	**▽5,0 ± 0,0**	**▽6,2 ± 0,9**	**▽4,3 ± 0,5**
wingNut	17	16,7 ± 2,6	**▽7,5 ± 1,3**	**▽8,8 ± 1,1**	**▽9,9 ± 1,9**
glass	55	29,2 ± 3,3	**▽11,7 ± 2,1**	**▽12,1 ± 2,4**	**▽10,5 ± 1,1**
iris	37	9,5 ± 0,6	**▽5,5 ± 0,6**	**▽4,5 ± 0,5**	**▽7,1 ± 1,2**
chowdary	15	5,0 ± 0,0	**▽4,0 ± 0,0**	**▽4,0 ± 0,0**	**▽3,9 ± 0,2**
dyrskjot	25	7,3 ± 0,4	**▽4,9 ± 0,6**	**▽5,2 ± 0,4**	**6,9 ± 1,2**
eTongueSugar	25	7,2 ± 1,7	**▽5,0 ± 0,9**	**6,7 ± 1,2**	**7,3 ± 1,5**
golub	49	14,8 ± 1,0	**▽10,5 ± 1,0**	**▽12,8 ± 0,8**	**15,4 ± 1,0**
gordon	15	11,0 ± 1,2	**▽8,4 ± 0,6**	**▽7,2 ± 1,0**	**11,0 ± 1,1**
leukemia	66	18,6 ± 1,7	**▽10,0 ± 1,9**	**▽9,6 ± 0,8**	**18,1 ± 1,8**
lung	40	18,2 ± 3,3	**▽10,9 ± 2,8**	**▽12,6 ± 1,8**	**17,4 ± 3,1**
miRNAcancer	218	78,6 ± 5,6	**▽27,2 ± 3,8**	**▽42,1 ± 5,8**	**▽56,7 ± 5,9**
Média	56,50	21,20	9,87	10,77	12,78
Redução em relação ao *Baseline*			53,43%	49,19%	39,71%

5.1.5 Resumo do Impacto na Qualidade e Número de Partições

Como diferentes pontos foram analisados a respeito do impacto das seleções na qualidade e no número de partições resultantes no MOCLE, um resumo se faz necessário. Considerada a metodologia experimental adotada, a seguir estão destacadas as principais observações relacionadas a cada uma das seleções:

Diversidade: Embora tenha sido a melhor em selecionar boas partições, ela impactou negativamente na qualidade dos resultados, se equiparando as piores seleções. Isso indica que o MOCLE, assim como outras técnicas de *ensemble* e de *multiple clustering*, não gera bons resultados considerando apenas diversidade em sua entrada.

SRD, SR e BRP: A análise aqui contida aponta que a causa do impacto negativo no MOCLE, em termos de qualidade, foi responsabilidade dessas seleções, por não selecionarem as melhores partições disponíveis. Isso porque os índices relativos considerados por essas seleções têm um forte viés para um tipo específico *cluster*, o que não vai de encontro as estruturas presentes em alguns *datasets* aqui utilizados. É possível que a utilização de outras combinações de índices relativos leve a maior flexibilização e melhores resultados. Também é possível que permitindo esses métodos selecionar um número maior de partições (além de 50% do tamanho de Π_{Rep}), esse quadro se reverta, porém, essas seleções provavelmente já estariam em desvantagem no número de partições geradas pelo MOCLE[4]. Em comparação com os outros métodos avaliados, esses foram os que mais impactaram negativamente no MOCLE.

HSS: Foi a seleção que menos forneceu partições ao MOCLE e, por isso, teve o maior risco de não selecionar partições boas. Porém, seu impacto na qualidade dos resultados do MOCLE foi mediano, quando comparado com as outras seleções.

FILTA: Impactou positivamente em termos de qualidade, sem grandes perdas em relação ao *baseline*, porém não auxiliou o MOCLE a produzir conjuntos de partições concisos. Além disso, essa seleção se mostrou mais sensível que o CAS em reduções no número de partições selecionadas.

ASA e CAS: Foram as duas seleções que mais beneficiaram o MOCLE; com o ASA tendo uma leve vantagem em diminuir o número de partições e o CAS uma leve vantagem em manter a qualidade dos resultados. Contudo, o ASA tem uma outra vantagem que vai ao encontro de um dos propósitos fundamentais do MOCLE, que é sua habilidade de inferir o número de partições a serem selecionadas. Mesmo

[4] E também no tempo de execução do MOCLE, já que seu tempo de execução é proporcional ao número de partições fornecidas a ele. Isso será melhor discutido na Seção 5.2

sendo esse parâmetro computado empiricamente, para os *datasets* testados, o ASA apresentou bons resultados[5].

[5] Alguns dos *datasets* aqui utilizados estavam presentes no trabalho de Sakata et al. (2010), onde o ASA foi empiricamente calibrado. Porém, aqui são utilizados algoritmos diferentes para gerar as partições e números diferentes de partições são obtidos. O que pode amenizar, se é que existem, quaisquer vieses.

5.2 Impacto no Tempo de Execução

Duas tabelas foram organizadas para a análise do impacto das seleções no tempo de execução do MOCLE. Na primeira, a Tabela 10, foi adotado o mesmo esquema de categorização e cores da Tabela 5, nela estão listados os tempos de execução do MOCLE. Por sua vez, na Tabela 11, estão listados os tempos de execução das seleções. Vale lembrar que todas essas medições foram realizadas no ambiente computacional descrito no Apêndice B.

Os dados presentes na Tabela 10 revelam que o tempo de execução do MOCLE é proporcional ao número de agrupamentos fornecidos como partições base (Π_I) — esses valores estão descritos na Tabela 7. Pois, todas as seleções que reduziram em 50% o número de partições, acabaram por reduzir em cerca de 50% o tempo de execução do MOCLE (em média). Contudo, é mais difícil perceber essa correlação nos casos do ASA e do HSS, pois ambas as seleções não são constantes no percentual de redução entre *datasets*, isto é, é possível reduzir muito um conjunto de partições para determinado *dataset* e reduzir menos em outro diferente. A Figura 9 comprova que para o ASA e o HSS essas flutuações também são proporcionais, logo, o tempo de execução do MOCLE é proporcional ao número de agrupamentos a ele fornecidos. Na Figura 9 estão listados o percentual de redução do Π_{Rep} e de tempo para cada *dataset* com respeito as seleções ASA e HSS. Essa figura também revela que o HSS tende a proporcionar reduções mais constantes de partições entre diferentes *datasets*, o que explica o motivo da discrepância entre as médias de redução de tempo e de número de partições ser maior para o ASA na Tabela 10.

Em suma, como a redução do tempo de execução do MOCLE é proporcional a redução no número de partições base (Π_I), as seleções que mais reduziram o tempo foram aquelas que menos selecionaram partições: o ASA e o HSS. É importante salientar que reduções no tempo de execução do MOCLE, iguais ao ASA e o HSS, podem ser obtidas pelas outras seleções, se elas forem parametrizadas para selecionar o mesmo número de partições que essas duas seleções.

A Tabela 11 revela variações nos tempos de execução para cada seleção, nela é possível observar que nenhuma seleção ultrapassou 52 segundos. Em comparação com os tempos de execução do MOCLE, utilizando todo Π_{Rep}, na Tabela 10, esses poucos segundos não fazem diferença, até mesmo nos menores *datasets*. É possível observar isso na Figura 10, em que constam as medições de tempo do HSS[6]. E também na Figura 11, em que constam as medições de tempo do SRD[7]. Para as outras seleções que selecionaram o mesmo número de partições que o SRD o comportamento foi muito semelhante, por isso os gráficos foram omitidos. O ASA teve um comportamento mais parecido com o HSS na

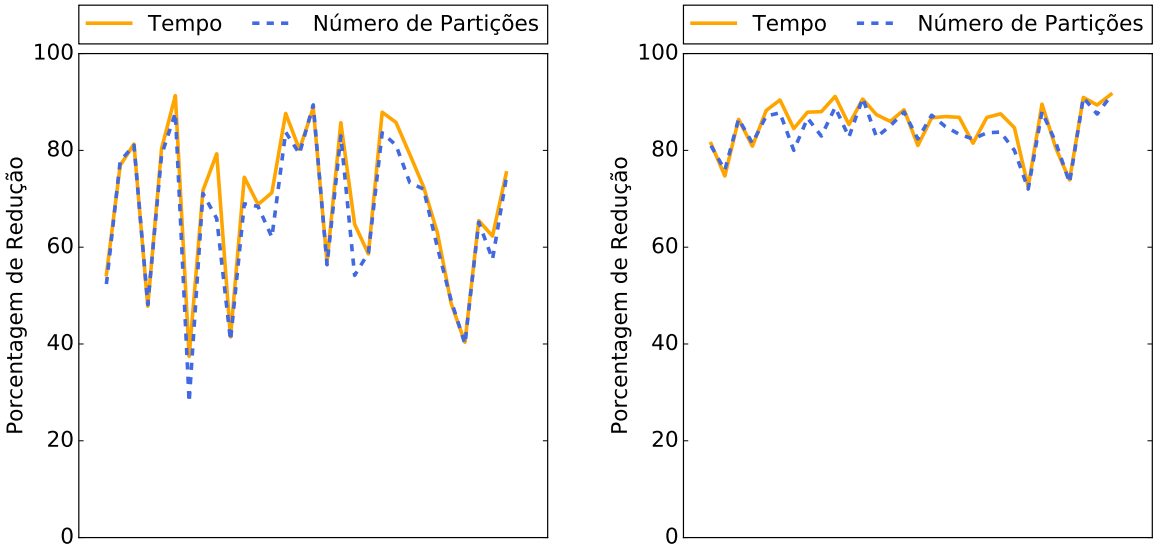
[6] O HSS foi a seleção que mais diminuiu o tempo de execução do MOCLE (em média), como a Tabela 10 demonstra

[7] O SRD foi a seleção mais lenta (em média), como consta na Tabela 11

Figura 12, já que reduziu mais partições que 50% de Π_{Rep} (na média).

Em suma, a Tabela 11 e as Figuras 10, 11 e 12 demonstram que o tempo de execução das seleções, independente de qual seja, é muito inferior ao tempo de execução do MOCLE. Assim, pode-se concluir que o tempo de execução total do MOCLE não é definido pelo algoritmo de seleção, mas sim, pela quantidade de partições que o algoritmo de seleção resulta.

Figura 9 – Proporções de redução do Π_{Rep} e de tempo para todas as bases de dados. Os nomes dos *datasets* foram removidos, pois aqui não são relevantes.



(a) Proporções de redução para o ASA.

(b) Proporções de redução para o HSS.

Fonte: elaboração própria.

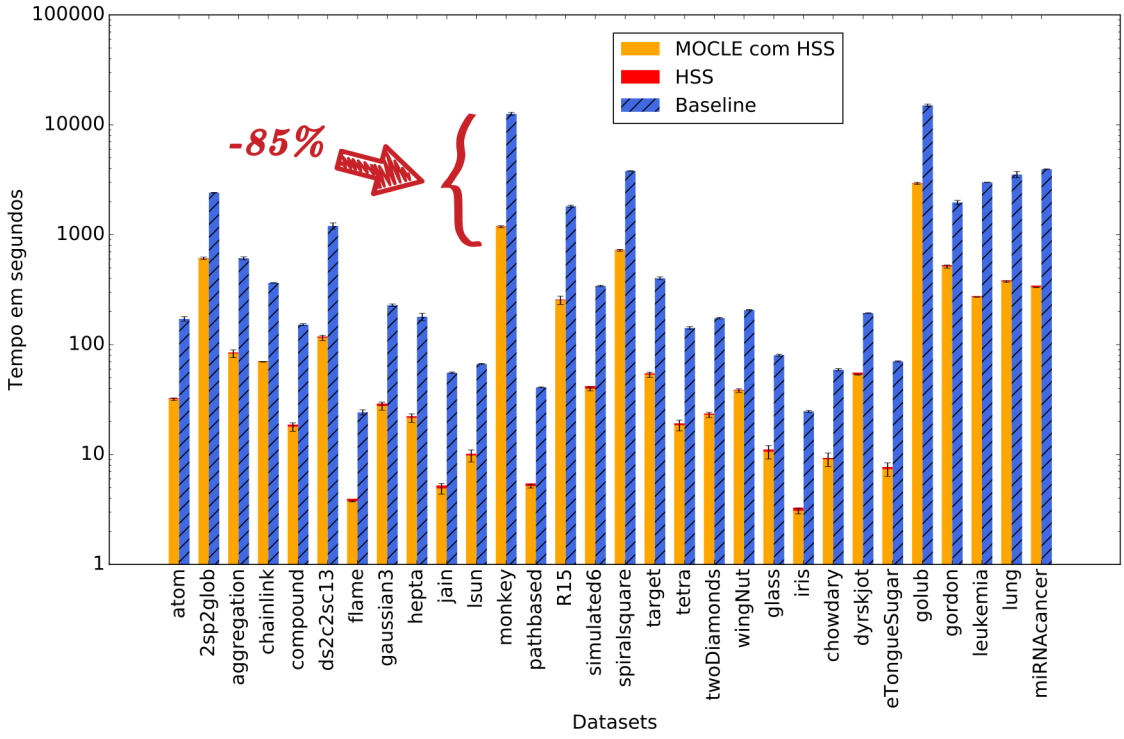
Tabela 10 – Médias e desvios padrão do tempo em segundos de execução resultantes das múltiplas execuções do MOCLE sob a influência das diferentes seleções.

Dataset	*Baseline*	SRD	SR	BRP	ASA	Diversidade	CAS	FILTA	HSS
atom	172 ± 7,9	▽**86 ± 1,8**	▽**86 ± 1,9**	▽**86 ± 1,4**	▽**78 ± 1,1**	▽**90 ± 4,3**	▽**91 ± 4,1**	▽**96 ± 0,6**	▽**32 ± 0,9**
2sp2glob	2413 ± 12,7	▽**1282 ± 23,7**	▽**1284 ± 15,3**	▽**1282 ± 22,3**	▽**558 ± 11,5**	▽**1226 ± 7,5**	▽**1243 ± 40,6**	▽**1258 ± 25,5**	▽**610 ± 14,8**
aggregation	611 ± 19,4	▽**320 ± 9,5**	▽**313 ± 6,8**	▽**292 ± 5,6**	▽**114 ± 3,6**	▽**302 ± 7,0**	▽**293 ± 7,0**	▽**342 ± 9,2**	▽**83 ± 6,8**
chainlink	365 ± 2,0	▽**190 ± 5,3**	▽**190 ± 0,9**	▽**191 ± 0,8**	▽**190 ± 0,9**	▽**191 ± 0,8**	▽**190 ± 2,8**	▽**191 ± 0,7**	▽**70 ± 0,5**
compound	152 ± 3,1	▽**73 ± 0,4**	▽**73 ± 0,4**	▽**73 ± 0,5**	▽**30 ± 0,6**	▽**74 ± 1,7**	▽**73 ± 2,1**	▽**76 ± 1,9**	▽**18 ± 1,6**
ds2c2sc13	1203 ± 79,1	▽**499 ± 22,9**	▽**522 ± 25,3**	▽**515 ± 26,0**	▽**104 ± 9,5**	▽**437 ± 29,0**	▽**599 ± 33,2**	▽**371 ± 6,7**	▽**115 ± 6,4**
flame	24 ± 1,3	▽**11 ± 1,0**	▽**10 ± 0,1**	▽**10 ± 0,1**	▽**15 ± 0,5**	▽**10 ± 0,3**	▽**10 ± 0,2**	▽**10 ± 0,1**	▽**4 ± 0,0**
gaussian3	228 ± 5,6	▽**127 ± 5,1**	▽**126 ± 5,1**	▽**116 ± 2,1**	▽**65 ± 2,8**	▽**100 ± 1,9**	▽**110 ± 2,4**	▽**108 ± 3,3**	▽**28 ± 2,4**
hepta	179 ± 13,8	▽**93 ± 5,9**	▽**90 ± 6,6**	▽**80 ± 5,3**	▽**37 ± 2,9**	▽**60 ± 2,6**	▽**75 ± 5,0**	▽**73 ± 5,2**	▽**22 ± 2,0**
jain	56 ± 0,8	▽**26 ± 0,9**	▽**27 ± 0,4**	▽**27 ± 0,2**	▽**33 ± 0,5**	▽**28 ± 0,3**	▽**28 ± 0,1**	▽**28 ± 0,9**	▽**5 ± 0,5**
lsun	67 ± 0,8	▽**35 ± 1,7**	▽**35 ± 1,5**	▽**35 ± 0,5**	▽**17 ± 1,8**	▽**32 ± 2,0**	▽**35 ± 1,9**	▽**34 ± 0,9**	▽**10 ± 1,2**
monkey	12582 ± 371,9	▽**6242 ± 108,5**	▽**6356 ± 132,2**	▽**6150 ± 166,1**	▽**3922 ± 64,1**	▽**6201 ± 63,6**	▽**6351 ± 115,5**	▽**6685 ± 154,9**	▽**1182 ± 24,0**
pathbased	41 ± 0,5	▽**18 ± 1,3**	▽**17 ± 0,1**	▽**17 ± 0,1**	▽**12 ± 0,4**	▽**17 ± 0,5**	▽**17 ± 0,2**	▽**17 ± 0,2**	▽**5 ± 0,2**
R15	1814 ± 42,5	▽**821 ± 12,2**	▽**850 ± 20,3**	▽**828 ± 13,0**	▽**224 ± 1,7**	▽**848 ± 41,7**	▽**915 ± 60,7**	▽**825 ± 17,0**	▽**254 ± 22,5**
simulated6	342 ± 5,1	▽**168 ± 1,6**	▽**169 ± 2,5**	▽**162 ± 6,1**	▽**68 ± 2,4**	▽**146 ± 3,0**	▽**164 ± 4,9**	▽**160 ± 5,0**	▽**40 ± 1,6**
spiralsquare	3797 ± 25,1	▽**1886 ± 11,2**	▽**1886 ± 12,1**	▽**1898 ± 15,0**	▽**425 ± 10,2**	▽**1969 ± 26,4**	▽**1901 ± 12,2**	▽**2013 ± 19,5**	▽**722 ± 13,0**
target	402 ± 11,9	▽**199 ± 8,1**	▽**200 ± 10,0**	▽**205 ± 6,1**	▽**173 ± 5,7**	▽**199 ± 5,0**	▽**207 ± 5,3**	▽**211 ± 5,0**	▽**53 ± 2,9**
tetra	142 ± 4,0	▽**70 ± 1,9**	▽**73 ± 0,3**	▽**71 ± 2,0**	▽**20 ± 1,3**	▽**67 ± 2,7**	▽**74 ± 1,1**	▽**75 ± 0,8**	▽**18 ± 2,1**
twoDiamonds	174 ± 2,7	▽**88 ± 3,9**	▽**88 ± 1,6**	▽**78 ± 4,1**	▽**61 ± 2,6**	▽**65 ± 2,0**	▽**67 ± 1,4**	▽**66 ± 1,3**	▽**23 ± 1,2**
wingNut	206 ± 2,1	▽**109 ± 0,7**	▽**110 ± 0,9**	▽**109 ± 0,9**	▽**85 ± 0,6**	▽**109 ± 1,0**	▽**110 ± 1,3**	▽**110 ± 0,9**	▽**38 ± 1,5**
glass	80 ± 1,6	▽**39 ± 0,8**	▽**39 ± 1,3**	▽**38 ± 1,1**	▽**10 ± 1,2**	▽**37 ± 2,4**	▽**41 ± 1,2**	▽**38 ± 1,0**	▽**11 ± 1,5**
iris	25 ± 0,5	▽**10 ± 0,9**	▽**9 ± 0,0**	▽**9 ± 0,1**	▽**4 ± 0,2**	▽**10 ± 0,1**	▽**9 ± 0,1**	▽**10 ± 0,1**	▽**3 ± 0,2**
chowdary	59 ± 1,4	▽**32 ± 0,5**	▽**32 ± 0,1**	▽**32 ± 0,4**	▽**12 ± 1,8**	▽**29 ± 2,6**	▽**32 ± 0,1**	▽**31 ± 0,9**	▽**9 ± 1,3**
dyrskjot	194 ± 1,6	▽**100 ± 0,5**	▽**100 ± 0,3**	▽**100 ± 0,4**	▽**53 ± 0,3**	▽**100 ± 0,6**	▽**100 ± 0,6**	▽**100 ± 0,5**	▽**53 ± 0,6**
eTongueSugar	70 ± 0,7	▽**37 ± 0,6**	▽**37 ± 0,5**	▽**37 ± 0,4**	▽**26 ± 1,2**	▽**36 ± 0,9**	▽**36 ± 0,4**	▽**36 ± 1,0**	▽**7 ± 1,0**
golub	15017 ± 400,8	▽**7661 ± 272,5**	▽**7777 ± 121,4**	▽**7695 ± 134,1**	▽**7751 ± 94,6**	▽**7635 ± 113,3**	▽**7988 ± 151,2**	▽**7946 ± 58,5**	▽**2929 ± 63,1**
gordon	1965 ± 88,0	▽**1036 ± 37,7**	▽**1059 ± 37,0**	▽**1053 ± 29,3**	▽**1172 ± 13,6**	▽**1042 ± 42,8**	▽**1059 ± 38,9**	▽**1059 ± 39,8**	▽**513 ± 17,0**
leukemia	2994 ± 8,2	▽**1488 ± 5,5**	▽**1484 ± 3,5**	▽**1486 ± 9,0**	▽**1033 ± 2,0**	▽**1486 ± 4,1**	▽**1489 ± 3,5**	▽**1482 ± 5,9**	▽**271 ± 0,6**
lung	3528 ± 231,6	▽**1685 ± 127,1**	▽**1571 ± 40,1**	▽**1558 ± 27,1**	▽**1331 ± 27,4**	▽**1528 ± 37,3**	▽**1557 ± 13,7**	▽**1534 ± 38,5**	▽**375 ± 6,6**
miRNAcancer	3944 ± 26,2	▽**1983 ± 10,5**	▽**1999 ± 13,2**	▽**2004 ± 11,3**	▽**972 ± 2,8**	▽**1887 ± 3,0**	▽**1982 ± 13,6**	▽**2052 ± 15,6**	▽**332 ± 1,3**
Média	1761,5	880,4	887,1	874,5	619,9	865,4	894,8	901,2	261,2
Redução em relação ao *Baseline*		50,02%	49,64%	50,35%	64,81%	50,87%	49,20%	48,84%	85,17%

Tabela 11 – Tempo em segundos de execução das seleções.

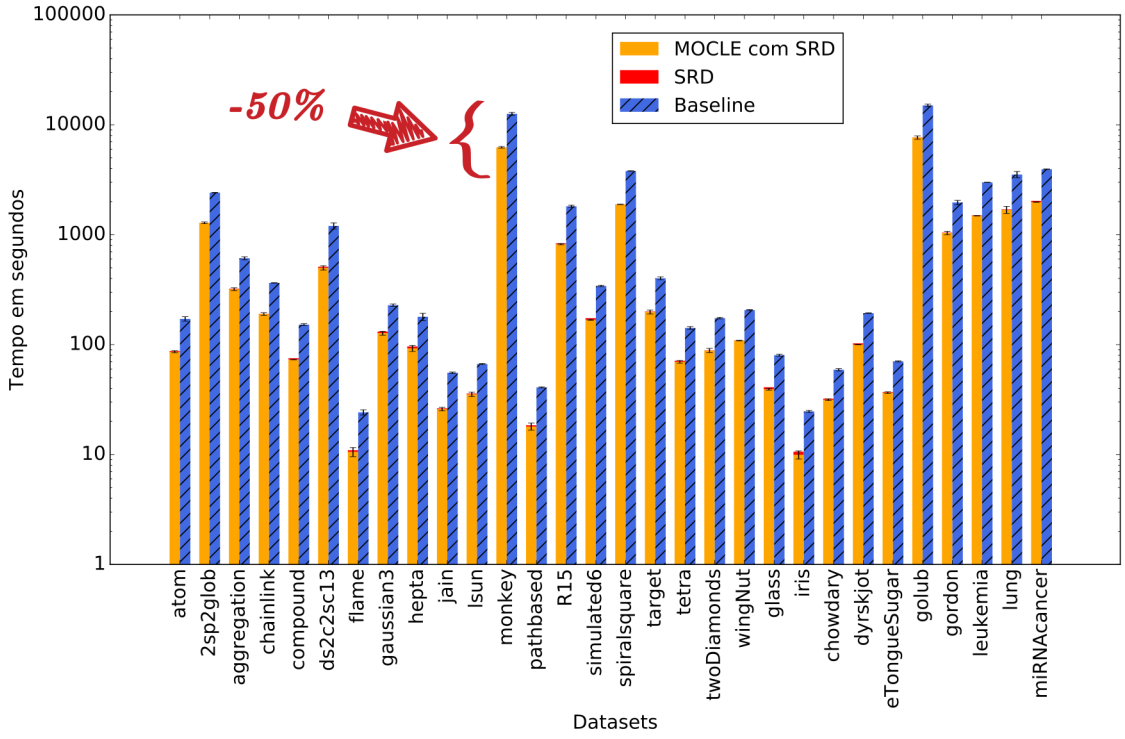
Dataset	SRD	SR	BRP	ASA	Diversidade	CAS	FILTA	HSS
atom	1,47	1,31	1,31	0,37	0,19	0,21	0,36	0,61
2sp2glob	2,61	1,5	1,49	1,99	1,26	1,29	2,74	3,8
aggregation	3,45	1,38	1,37	2,88	1,53	1,57	1,93	2,01
chainlink	0,76	0,5	0,5	0,54	0,31	0,32	0,58	0,92
compound	2,1	0,75	0,74	1,75	0,93	0,96	1,12	0,87
ds2c2sc13	12,46	3,29	3,26	11,03	5,68	5,76	6,51	3,3
flame	0,38	0,18	0,18	0,37	0,16	0,17	0,2	0,21
gaussian3	4,29	3,64	3,63	0,85	0,44	0,47	0,52	1,49
hepta	3,73	0,95	0,94	3,31	1,66	1,71	1,93	0,82
jain	0,44	0,2	0,2	0,42	0,2	0,23	0,28	0,33
lsun	0,55	0,27	0,27	0,49	0,24	0,26	0,31	0,39
monkey	6,07	3,11	3,1	5,86	3,6	3,63	11,72	12,84
pathbased	0,48	0,2	0,2	0,53	0,22	0,24	0,28	0,31
R15	14,31	3,76	3,73	12,86	6,6	6,69	7,54	3,65
simulated6	6,11	4,99	4,98	1,44	0,72	0,74	0,84	2,29
spiralsquare	4,81	1,82	1,81	4,57	2,69	2,73	5,1	6,16
target	1,82	0,7	0,7	1,69	0,9	0,93	1,35	1,59
tetra	1,51	0,54	0,54	1,39	0,69	0,71	0,84	0,76
twoDiamonds	0,46	0,26	0,26	0,41	0,23	0,24	0,4	0,65
wingNut	0,31	0,2	0,2	0,29	0,16	0,17	0,32	0,59
glass	1,63	0,6	0,59	1,4	0,67	0,7	0,81	0,56
iris	0,67	0,23	0,22	0,61	0,3	0,32	0,36	0,22
chowdary	0,74	0,67	0,67	0,14	0,08	0,09	0,09	0,27
dyrskjot	2,77	2,56	2,56	0,45	0,23	0,24	0,26	2,39
eTongueSugar	0,46	0,25	0,25	0,35	0,18	0,2	0,25	0,36
golub	24,18	23,39	23,39	1,43	0,79	0,81	0,93	30,55
gordon	8,57	8,49	8,49	0,49	0,36	0,37	0,4	15,69
leukemia	12,95	11,4	11,39	2,2	1,09	1,13	1,36	5,52
lung	16,47	15,94	15,94	0,99	0,53	0,55	0,65	9,84
miRNAcancer	51,07	27,83	27,75	26,73	13,22	13,36	14,81	12,15
Média	6,56	4,22	4,21	3,05	1,59	1,62	2,20	4,17

Figura 10 – Comparação do tempo de execução do *baseline* em relação ao MOCLE em associação com o HSS mais o tempo de execução do HSS. Gráfico em escala logarítmica.



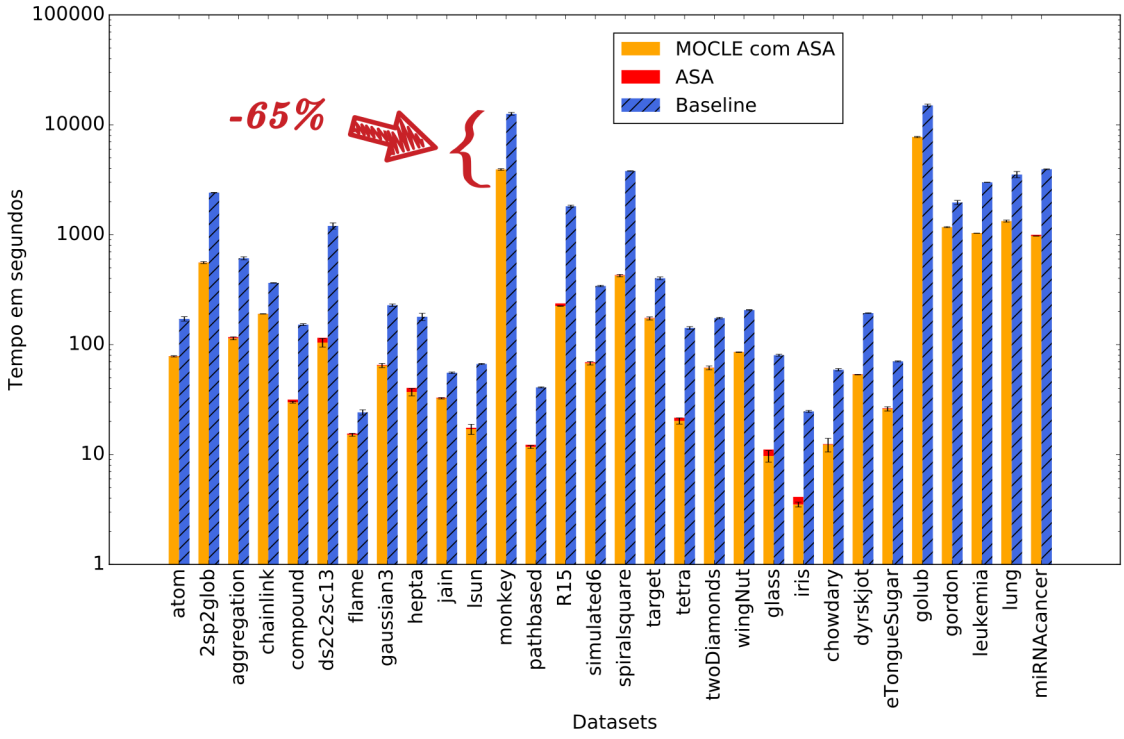
Fonte: elaboração própria.

Figura 11 – Comparação do tempo de execução do *baseline* em relação ao MOCLE em associação com o SRD mais o tempo de execução do SRD. Gráfico em escala logarítmica.



Fonte: elaboração própria.

Figura 12 – Comparação do tempo de execução do *baseline* em relação ao MOCLE em associação com o ASA mais o tempo de execução do ASA. Gráfico em escala logarítmica.



Fonte: elaboração própria.

5.3 Considerações sobre os Resultados

Avaliando os três critérios propostos para a comparação entre as seleções — impacto na qualidade, número de partições e tempo de execução —, três delas se destacam: o ASA, o CAS e o HSS. O HSS, por sua habilidade em reduzir o número de partições e o tempo de execução sem prejuízos consideráveis em qualidade. O ASA e o CAS, pelo fato de manterem a qualidade dos resultados do MOCLE estável, reduzindo os outros dois fatores (tempo de execução e número de partições). Porém, o ASA se destaca ainda mais por sua capacidade em inferir automaticamente o número de partições a serem selecionadas, o que vai de encontro a um dos propósitos fundamentais do MOCLE, ser uma ferramenta com poucos parâmetros. O HSS também possui essa capacidade, o que o tornaria útil em cenários onde o tempo é o fator mais importante, já que o ASA apresentou melhores resultados nos outros dois fatores analisados (qualidade e número de partições).

Com os experimentos aqui realizados ainda é difícil apontar os reais motivos do desempenho superior do ASA e do CAS em termos de qualidade, mesmo com reduções maiores no número de partições selecionadas. Contudo, uma possível explicação para essa melhora, é que ambas as seleções ao utilizarem apenas um índice externo de validação, o ARI, para medir a qualidade e a diversidade das partições, introduziram menos vieses no conjunto de partições selecionadas. E, por isso, se saíram melhor em fornecer partições relevantes ao MOCLE na maioria dos casos.

Sendo assim, considerando nossa metodologia experimental, observando os resultados obtidos e as características do algoritmo, o ASA foi o método de seleção que melhor funcionou em associação com o MOCLE. Com ele, foi possível atingir os objetivos propostos para esse trabalho. E, ao que tudo indica, é um mecanismo de grande valor para ser adicionado no *framework* MOCLE. Este pode ser um indicativo de que, a seleção de partições, pode ser benéfica para a área de *ensemble* multiobjetivo.

Conclusão

No presente trabalho, foram conduzidos experimentos para avaliar se o MOCLE, uma importante técnica de agrupamento, também se beneficiaria do pré-processamento das suas múltiplas partições de entrada, assim como outros trabalhos de *multiple clustering* (FERN; LIN, 2008; NALDI; CARVALHO; CAMPELLO, 2013; LEI et al., 2016). Na presente análise, foram utilizados três critérios, oriundos de três oportunidades de melhora identificadas em experimentos preliminares (PEDOTE; FACELI; SAKATA, 2017). Sendo eles: qualidade dos resultados, número de partições que compõem a saída e tempo de execução.

Para pré-processar as partições de entrada da fase de otimização do MOCLE, diversos métodos de seleção de partições disponíveis em diversas subáreas da literatura de agrupamento foram avaliados. Desses métodos, nos três critérios avaliados, nenhum demonstrou ter forte impacto negativo sobre o funcionamento do MOCLE e seus resultados. Pelo contrário, alguns métodos se sobressaíram e melhoraram, em boa medida, o desempenho do MOCLE. Em particular, o CAS de Fern e Lin (2008), o ASA de Sakata et al. (2010) e o HSS de Antunes, Faceli e Sakata (2017).

Resultados experimentais mostraram que ASA e o CAS foram as melhores seleções considerando todos os três fatores analisados. Já o HSS, sem grandes perdas de qualidade, auxiliou o MOCLE a reduzir consideravelmente seu tempo de execução, que é um fator limitante na aplicação do mesmo (FACELI; CARVALHO; SOUTO, 2008). Além disso, o ASA e HSS têm a vantagem de conseguirem inferir o número de partições a serem selecionadas automaticamente, resultando em uma necessidade menor de conhecimento por parte do usuário/especialista para calibrar parâmetros. Isso, vai de encontro a um dos princípios do funcionamento do MOCLE, que é realizar importantes passos da análise de agrupamento de forma automática, com o menor número de parâmetros possível — com o intuito de produzir bons resultados, mesmo que o usuário não seja especialista ou tenha grande conhecimento de análise de agrupamentos.

Trabalhos Futuros e Limitações

A seguir está uma listagem de possível trabalhos futuros. Alguns desses itens são relacionados a mitigar as limitações da presente análise e outros se referem à ampliação da mesma.

- A primeira possibilidade é analisar a influência dessas seleções no processamento do MOCLE em *datasets* oriundos de outras áreas do conhecimento, já que nesse trabalho

a grande maioria dos *datasets* reais tiveram origem em problemas de bioinformática[8]. Outra alternativa nessa mesma linha de raciocínio é utilizar *datasets* artificiais com o mesmo propósito, porém em situações controladas.

- Outra alternativa é elaborar mais experimentos ou análises para explicar melhor o(s) motivo(s) do ASA e do CAS impactarem positivamente nos resultados do MOCLE; com o intuito de aprimorar o *framework* e também as seleções.

- Outro trabalho possível é estender o MOCLE adicionando de forma definitiva o ASA como um passo de pré-processamento das partições.

- Outra possibilidade interessante é avaliar o desempenho utilizando, dentro das seleções, um outro índice externo, além do ARI. Pois, há evidências de que o ARI talvez não seja a melhor medida para o uso aqui dado, pelas razões discutidas em Souto et al. (2012).

- Para melhorar o desempenho das seleções CRI (i.e., SR, SRD e BRP) em situações semelhantes as aqui experimentalmente retratadas, também é possível avaliá-las com outros índices relativos que apresentem comportamento mais abrangente, favorecendo também partições com *clusters* de diferentes formatos.

- Outro caminho relevante para melhora do desempenho do MOCLE, porém fora da linha de investigação desse trabalho, é a possibilidade de alterar o *framework* para incorporar técnicas de paralelização e/ou distribuição.

- Por fim, outra possibilidade é avaliar o impacto desses métodos de seleção em outras técnicas de *ensemble* multiobjetivo, como o *Multi-objective Clustering with Hierarchical Partitions Fusions* (MCHPF) (COELHO; FERNANDES; FACELI, 2010).

[8] A única exceção é o *dataset* `eTongueSugar`, que é relacionado a área de química.

Referências

ANTUNES, V. *Estratégia Híbrida de Seleção de Partições para o Problema de Agrupamento de Dados*. Dissertação (Mestrado) — Universidade Federal de São Carlos - Campus Sorocaba, UFSCar, Sorocaba, São Paulo - Brasil, 2018. Citado 4 vezes nas páginas 4, 7, 22 e 23.

ANTUNES, V.; FACELI, K.; SAKATA, T. C. HSS: Compact set of Partitions via Hybrid Selection. In: *Procedings of the 2017 VI Brazilian Conference on Intelligent Systems*. Uberlândia, MG, Brazil: IEEE Computer Society, 2017. (BRACIS '17), p. 37–42. Citado 4 vezes nas páginas 6, 7, 22 e 57.

BAILEY, J. Alternative Clustering Analysis: A Review. In: AGGARWAL, C. C.; REDDY, C. K. (Ed.). *Data Clustering: Algorithms and Applications*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2013. p. 533–548. ISBN 1-4665-5821-0 978-1-4665-5821-2. Citado na página 1.

BEZDEK, J. C.; PAL, N. R. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 28, n. 3, p. 301–315, jun. 1998. ISSN 1083-4419. Citado na página 69.

CALIńSKI, T.; HARABASZ, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, v. 3, n. 1, p. 1–27, 1974. Citado na página 67.

CARUANA, R. et al. Meta Clustering. In: *Proceedings of the Sixth International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2006. (ICDM '06), p. 107–118. ISBN 0-7695-2701-9. Disponível em: <http://dx.doi.org/10.1109/ICDM.2006.103>. Citado na página 21.

COELHO, A. L. V.; FERNANDES, E.; FACELI, K. Letters: Inducing Multi-objective Clustering Ensembles with Genetic Programming. *Neurocomput.*, v. 74, n. 1-3, p. 494–498, dez. 2010. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2010.09.014>. Citado na página 58.

DAVIES, D. L.; BOULDIN, D. W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 1, n. 2, p. 224–227, fev. 1979. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.1979.4766909>. Citado na página 68.

DEMšAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006. Citado na página 30.

DUNN, J. C. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, v. 4, n. 1, p. 95–104, 1974. Citado na página 69.

FACELI, K. *Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento*. Tese (Doutorado) — Universidade de São Paulo, USP, São Carlos, São Paulo - Brasil, 2006. Disponível em: <http://dx.doi.org/10.11606/T.55.2006.tde-12012007-082216>. Citado 10 vezes nas páginas 5, 9, 10, 25, 27, 28, 29, 30, 32 e 38.

FACELI, K.; CARVALHO, A.; SOUTO, M. C. de. Cluster ensemble and multi-objective clustering methods. *Pattern Recognition Technologies and Applications: Recent Advances*, 2008. Disponível em: <http://www.igi-global.com/viewtitle.aspx?TitleId=28037>. Citado 7 vezes nas páginas 4, 5, 9, 10, 25, 32 e 57.

FACELI, K.; CARVALHO, A. C. P. L. F. de; SOUTO, M. C. P. de. Multi-Objective Clustering Ensemble. In: *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*. Washington, DC, USA: IEEE Computer Society, 2006. (HIS '06), p. 51–. ISBN 0-7695-2662-4. Disponível em: <http://dx.doi.org/10.1109/HIS.2006.49>. Citado 2 vezes nas páginas 5 e 9.

FACELI, K.; CARVALHO, A. C. P. L. F. de; SOUTO, M. C. P. de. Multi-objective Clustering Ensemble. *Int. J. Hybrid Intell. Syst.*, v. 4, n. 3, p. 145–156, ago. 2007. ISSN 1448-5869. Disponível em: <http://dl.acm.org/citation.cfm?id=1367012.1367014>. Citado 2 vezes nas páginas 5 e 9.

FACELI, K.; CARVALHO, A. C. P. L. F. de; SOUTO, M. C. P. de. Multi-objective Clustering Ensemble with Prior Knowledge. In: *Proceedings of the 2Nd Brazilian Conference on Advances in Bioinformatics and Computational Biology*. Berlin, Heidelberg: Springer-Verlag, 2007. (BSB'07), p. 34–45. ISBN 3-540-73730-8 978-3-540-73730-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1776474.1776479>. Citado 2 vezes nas páginas 5 e 9.

FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro, RJ, Brasil: LTC - Livros Técnicos e Científicos, 2011. ISBN 978-85-216-1880-5. Citado 10 vezes nas páginas 1, 2, 3, 4, 9, 10, 14, 65, 66 e 70.

FACELI, K.; SAKATA, T. C. *Multiple solutions in cluster analysis: partitions x clusters*. Sorocaba, SP, Brazil, 2016. Disponível em: <http://lasid.sor.ufscar.br/clustersEvaluationBenchmark/>. Citado na página 26.

FACELI, K. et al. Partitions Selection Strategy for Set of Clustering Solutions. *Neurocomput.*, v. 73, n. 16-18, p. 2809–2819, out. 2010. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2010.03.028>. Citado na página 28.

FACELI, K.; SOUTO, M. C. P.; CARVALHO, A. C. P. L. F. d. A Strategy for the Selection of Solutions of the Pareto Front Approximation in Multi-objective Clustering Approaches. In: *Proceedings of the 2008 10th Brazilian Symposium on Neural Networks*. Washington, DC, USA: IEEE Computer Society, 2008. (SBRN '08), p. 27–32. ISBN 978-0-7695-3361-2. Disponível em: <http://dx.doi.org/10.1109/SBRN.2008.34>. Citado na página 15.

FACELI, K. et al. Multi-objective Clustering Ensemble for Gene Expression Data Analysis. *Neurocomput.*, v. 72, n. 13-15, p. 2763–2774, ago. 2009. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2008.09.025>. Citado 5 vezes nas páginas 5, 9, 10, 11 e 32.

FACELI, K.; SOUTO, M. D.; CARVALHO, A. D. Multi-objective Clustering Ensemble: A Framework for Cluster Analysis. *International Journal of Soft Computing and Bioinformatics*, v. 1, p. 9–17, 2010. Citado 2 vezes nas páginas 5 e 9.

FAIVISHEVSKY, L.; GOLDBERGER, J. A Nonparametric Information Theoretic Clustering Algorithm. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. USA: Omnipress, 2010. (ICML'10), p. 351–358. ISBN 978-1-60558-907-7. Disponível em: <http://dl.acm.org/citation.cfm?id=3104322.3104368>. Citado na página 21.

FERN, X. Z.; LIN, W. Cluster ensemble selection. *Statistical Analysis and Data Mining*, v. 1, n. 3, p. 128–141, 2008. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1002/sam.10008/full>. Citado 9 vezes nas páginas 6, 7, 13, 14, 17, 20, 21, 39 e 57.

FRÄNTI, P. *Clustering datasets*. 2015. Disponível em: <http://cs.uef.fi/sipu/datasets/>. Citado na página 26.

GUERRA, L. et al. A Comparison of Clustering Quality Indices Using Outliers and Noise. *Intell. Data Anal.*, v. 16, n. 4, p. 703–715, jul. 2012. ISSN 1088-467X. Disponível em: <http://dx.doi.org/10.3233/IDA-2012-0545>. Citado na página 65.

HADJITODOROV, S. T.; KUNCHEVA, L. I.; TODOROVA, L. P. Moderate Diversity for Better Cluster Ensembles. *Inf. Fusion*, v. 7, n. 3, p. 264–275, set. 2006. ISSN 1566-2535. Disponível em: <http://dx.doi.org/10.1016/j.inffus.2005.01.008>. Citado na página 39.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Clustering Validity Checking Methods: Part II. *SIGMOD Rec.*, v. 31, n. 3, p. 19–27, set. 2002. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/601858.601862>. Citado 2 vezes nas páginas 69 e 70.

HANDL, J.; KNOWLES, J. Multiobjective clustering with automatic determination of the number of clusters. *Technical Report*, 2004. Citado na página 5.

HANDL, J.; KNOWLES, J. An Evolutionary Approach to Multiobjective Clustering. *IEEE Transactions on Evolutionary Computation*, v. 11, n. 1, p. 56–76, fev. 2007. ISSN 1089-778X. Citado 2 vezes nas páginas 3 e 38.

HRUSCHKA, E. R.; CAMPELLO, R. J. G. B.; CASTRO, L. N. de. Evolving Clusters in Gene-expression Data. *Inf. Sci.*, v. 176, n. 13, p. 1898–1927, jul. 2006. ISSN 0020-0255. Disponível em: <http://dx.doi.org/10.1016/j.ins.2005.07.015>. Citado na página 67.

HRUSCHKA, E. R.; CASTRO, L. N. d.; CAMPELLO, R. J. G. B. Evolutionary Algorithms for Clustering Gene-Expression Data. In: *Proceedings of the Fourth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2004. (ICDM '04), p. 403–406. ISBN 0-7695-2142-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1032649.1033491>. Citado na página 66.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, v. 2, n. 1, p. 193–218, 1985. Disponível em: <http://link.springer.com/article/10.1007/BF01908075>. Citado 2 vezes nas páginas 14 e 65.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 72.

JAIN, A. K. Data Clustering: 50 Years Beyond K-means. *Pattern Recogn. Lett.*, v. 31, n. 8, p. 651–666, jun. 2010. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2009.09.011>. Citado 2 vezes nas páginas 1 e 3.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X. Citado na página 1.

JASZKIEWICZ, A. Momhlib++: Multiple objective metaheuristics library in c++. *URL: http://www-idss. cs. put. poznan. pl/˜ jaszkiewicz/MOMHLib abgerufen am*, v. 22, 2005. Citado na página 71.

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001. Disponível em: <http://www.scipy.org/>. Citado na página 72.

KARYPIS, G.; KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 20, n. 1, p. 359–392, dez. 1998. ISSN 1064-8275. Disponível em: <http://dx.doi.org/10.1137/S1064827595287997>. Citado na página 71.

LEI, Y. et al. FILTA: Better View Discovery from Collections of Clusterings via Filtering. In: CALDERS, T. et al. (Ed.). *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 145–160. ISBN 978-3-662-44851-9. Disponível em: <http://dx.doi.org/10.1007/978-3-662-44851-9_10>. Citado 3 vezes nas páginas 6, 7 e 21.

LEI, Y. et al. rFILTA: relevant and nonredundant view discovery from collections of clusterings via filtering and ranking. *Knowledge and Information Systems*, p. 1–41, 2016. ISSN 0219-3116. Disponível em: <http://dx.doi.org/10.1007/s10115-016-1008-y>. Citado 4 vezes nas páginas 6, 7, 21 e 57.

MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56. Citado na página 72.

MULLER, E. et al. Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data. In: *Proceedings of the 2010 IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2010. (ICDM '10), p. 1220–. ISBN 978-0-7695-4256-0. Disponível em: <http://dx.doi.org/10.1109/ICDM.2010.85>. Citado 5 vezes nas páginas 1, 2, 6, 21 e 39.

NALDI, M. C. *Técnicas de combinação para agrupamento centralizado e distribuído de dados*. Tese (Doutorado) — Universidade de São Paulo, USP, São Carlos, São Paulo - Brasil, 2011. Citado 2 vezes nas páginas 66 e 68.

NALDI, M. C.; CARVALHO, A. C. P. L. F. de; CAMPELLO, R. J. G. B. Cluster Ensemble Selection Based on Relative Validity Indexes. *Data Min. Knowl. Discov.*, v. 27, n. 2, p. 259–289, set. 2013. ISSN 1384-5810. Disponível em: <http://dx.doi.org/10.1007/s10618-012-0290-x>. Citado 9 vezes nas páginas 6, 17, 18, 19, 39, 57, 66, 69 e 70.

NG, A. Y.; JORDAN, M. I.; WEISS, Y. On Spectral Clustering: Analysis and an Algorithm. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, USA: MIT Press, 2001. (NIPS'01), p. 849–856. Disponível em: <http://dl.acm.org/citation.cfm?id=2980539.2980649>. Citado na página 15.

PAKHIRA, M. K.; BANDYOPADHYAY, S.; MAULIK, U. Validity index for crisp and fuzzy clusters. *Pattern recognition*, v. 37, n. 3, p. 487–501, 2004. Citado na página 68.

PEDOTE, G. L.; FACELI, K.; SAKATA, T. C. Impact of Base Partition Selection on Multi-Objective Clustering Ensemble. In: *Procedings of the 2017 XIV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*. Uberlândia, MG, Brazil: [s.n.], 2017. (ENIAC '17), p. 948–959. Citado 6 vezes nas páginas 5, 6, 7, 18, 32 e 57.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 72.

PIANTONI, J. et al. Impact of Base Partitions on Multi-objective and Traditional Ensemble Clustering Algorithms. In: ARIK, S. et al. (Ed.). *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I*. Cham: Springer International Publishing, 2015. p. 696–704. ISBN 978-3-319-26532-2. DOI: 10.1007/978-3-319-26532-2_77. Disponível em: <https://doi.org/10.1007/978-3-319-26532-2_77>. Citado 2 vezes nas páginas 5 e 9.

POHLERT, T. *The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)*. [S.l.], 2014. R package. Disponível em: <https://CRAN.R-project.org/package=PMCMR>. Citado na página 72.

RASTIN, P.; KANAWATI, R. A Multiplex-network Based Approach for Clustering Ensemble Selection. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. New York, NY, USA: ACM, 2015. (ASONAM '15), p. 1332–1339. ISBN 978-1-4503-3854-7. Disponível em: <http://doi.acm.org/10.1145/2808797.2808825>. Citado na página 13.

ROUSSEEUW, P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.*, v. 20, n. 1, p. 53–65, nov. 1987. ISSN 0377-0427. Disponível em: <http://dx.doi.org/10.1016/0377-0427(87)90125-7>. Citado na página 66.

SAKATA, T. C. et al. Improvements in the Partitions Selection Strategy for Set of Clustering Solutions. In: *Proceedings of the 2010 Eleventh Brazilian Symposium on Neural Networks*. Washington, DC, USA: IEEE Computer Society, 2010. (SBRN '10), p. 49–54. ISBN 978-0-7695-4210-2. Disponível em: <http://dx.doi.org/10.1109/SBRN.2010.17>. Citado 7 vezes nas páginas 6, 7, 15, 16, 22, 48 e 57.

SCHUBERT, E. et al. A framework for clustering uncertain data. *PVLDB*, v. 8, n. 12, p. 1976–1979, 2015. Disponível em: <http://www.vldb.org/pvldb/vol8/p1976-schubert.pdf>. Citado na página 72.

SNYDER, P. tmpfs: A virtual memory file system. In: *In Proceedings of the Autumn 1990 European UNIX Users' Group Conference*. [S.l.: s.n.], 1990. p. 241–248. Citado na página 71.

SOUTO, M. C. P. de et al. A Comparison of External Clustering Evaluation Indices in the Context of Imbalanced Data Sets. In: *Proceedings of the 2012 Brazilian Symposium on Neural Networks*. Washington, DC, USA: IEEE Computer Society, 2012. (SBRN '12), p. 49–54. ISBN 978-0-7695-4823-4. Disponível em: <http://dx.doi.org/10.1109/SBRN.2012.25>. Citado 2 vezes nas páginas 30 e 58.

STREHL, A.; GHOSH, J. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 583–617, mar. 2003. ISSN 1532-4435. Disponível em: <https://doi.org/10.1162/153244303321897735>. Citado na página 14.

STREHL, A.; GHOSH, J. Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.*, v. 3, p. 583–617, mar. 2003. ISSN 1532-4435. Disponível em: <http://dx.doi.org/10.1162/153244303321897735>. Citado na página 28.

VEENMAN, C. J.; REINDERS, M. J. T.; BACKER, E. A Maximum Variance Cluster Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 24, n. 9, p. 1273–1280, set. 2002. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2002.1033218>. Citado na página 32.

VENDRAMIN, L.; CAMPELLO, R. J. G. B.; HRUSCHKA, E. R. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, p. n/a–n/a, jun. 2010. ISSN 19321864, 19321872. Disponível em: <http://doi.wiley.com/10.1002/sam.10080>. Citado 2 vezes nas páginas 18 e 67.

WALT, S. v. d.; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. *Computing in Science and Engg.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 13, n. 2, p. 22–30, mar. 2011. ISSN 1521-9615. Disponível em: <http://dx.doi.org/10.1109/MCSE.2011.37>. Citado na página 72.

ZHANG, Y.; LI, T. Extending Consensus Clustering to Explore Multiple Clustering Views. In: *SDM*. Mesa, Arizona, USA: Bing Liu , Huan Liu , Chris Clifton , Takashi Washio and Chandrika Kamath, 2011. p. 920–931. ISBN 978-0-89871-992-5. Citado na página 21.

ZIMEK, A.; VREEKEN, J. The Blind Men and the Elephant: On Meeting the Problem of Multiple Truths in Data from Clustering and Pattern Mining Perspectives. *Mach. Learn.*, v. 98, n. 1-2, p. 121–155, jan. 2015. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1007/s10994-013-5334-y>. Citado 2 vezes nas páginas 3 e 6.

APÊNDICE A – Avaliação de Agrupamentos

Índices de validação são estatísticas pelas quais a validade de um agrupamento é testada (FACELI et al., 2011). Eles podem ser divididos em três categorias: externos, internos e relativos. Índices externos medem a similaridade entre dois agrupamentos de um mesmo conjunto de dados. Normalmente, esses índices são utilizados para avaliar um agrupamento de acordo com uma estrutura previamente conhecida. Os índices internos, por sua vez, medem a qualidade de um agrupamento com base apenas nos dados dos quais ele provém, isto é, eles avaliam o ajuste entre uma partição gerada e os dados em questão. Já os índices relativos têm a função de comparar diferentes agrupamentos de acordo com algum aspecto (estabilidade ou adequação dos agrupamentos aos dados, como cita Faceli et al. (2011), por exemplo).

A seguir, nas seções A.1 e A.2, são descritos os índices externos e relativos de interesse para esse trabalho.

A.1 Índices externos

A.1.1 *Adjusted Rand Index* (ARI)

O *Adjusted Rand Index* (ARI), proposto por Hubert e Arabie (1985), compara duas partições e resulta em uma estatística que varia no intervalo $[-1, 1]$, sendo que valores menores ou próximos a 0 indicam que a semelhança entre as partições é dada pelo acaso e o valor 1 indica que ambas as partições são idênticas. Segundo Faceli et al. (2011), esse índice é um dos mais utilizados para validação externa em agrupamentos.

Uma forma de demonstrar como é feito o cálculo do ARI, baseada na abordagem de Guerra et al. (2012), é apresentada a seguir. Considere que $\mathcal{S}$ e $\mathcal{T}$ são as duas partições a serem comparadas e n o número de objetos contidos em cada uma delas. Assuma a como o número de pares de objetos que estão no mesmo *cluster* em $\mathcal{S}$ e $\mathcal{T}$, b o número de pares de objetos que estão no mesmo *cluster* em $\mathcal{S}$ mas não em $\mathcal{T}$, c o número de pares de objetos no mesmo *cluster* em $\mathcal{T}$ mas não em $\mathcal{S}$, e d o número de pares de objetos em diferentes *clusters* em ambas as partições $\mathcal{S}$ e $\mathcal{T}$. Então, o cálculo do ARI se dá pela

seguinte equação:

$$ARI(\mathcal{S},\mathcal{T}) = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d))]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \tag{A.1}$$

A.2 Índices internos relativos

A.2.1 Silhueta Simplificada (SS)

Índices do tipo silhueta avaliam um agrupamento com base na qualidade individual de cada *cluster* (proximidade entre seus objetos, i.e., compactação) e também na atribuição adequada de um objeto ao seu *cluster* (distância desse objeto ao *cluster* mais próximo, i.e., separação) (FACELI et al., 2011; NALDI, 2011). A Silhueta Simplificada (SS), apresentada em Hruschka, Castro e Campello (2004), é uma simplificação do cálculo original apresentado em Rousseeuw (1987). O índice SS, utiliza a distância entre os objetos e os centroides de seus respectivos *clusters* ao invés da distância entre todos os objetos (como é feito o cálculo original de Rousseeuw (1987)). O cômputo do SS é dado pelas Equações A.2 e A.3. Nelas, x_j é o j-ésimo objeto de um *dataset* que pertence a um *cluster* $C_p \in \{C_1 \dots C_k\}$, em que k é o número de *clusters* de uma dada partição. Além disso, a dissimilaridade do j-ésimo objeto e o centroide do *cluster* do qual ele pertence, C_p, é indicada por $a_{p,j}$. Já $b_{p,j}$ indica a dissimilaridade entre o j-ésimo objeto e o centroide do seu *cluster* vizinho mais próximo. Logo, a silhueta simplificada do objeto x_j é dada por:

$$s_{x_j} = \frac{b_{p,j} - a_{p,j}}{max\{a_{p,j}, b_{p,j}\}} \tag{A.2}$$

É válido notar que o denominador é apenas um termo para normalização e também que quanto maior o valor de s_{x_j}, melhor a atribuição do objeto x_j ao *cluster* C_p. Fora isso, se C_p é um *cluster* constituído apenas por um objeto, então $s_{x_j} = 0$ é assumido por convenção (NALDI; CARVALHO; CAMPELLO, 2013). Isso evita que o SS, que é dado pela Equação A.3, eleja uma solução trivial $k = n$ (com cada objeto no *dataset* formando um *cluster*) como a melhor. Segundo Naldi, Carvalho e Campello (2013), a melhor partição é aquela que maximiza SS, isso implica minimizar a distância intra-*cluster* ($a_{p,j}$) enquanto a distância entre os *clusters* ($b_{p,j}$) é maximizada.

$$SS = \frac{1}{n}\sum_{j=1}^{n} s_{x_j} \tag{A.3}$$

A.2.2 Silhueta Simplificada Alternativa (SSA)

Uma variante da Silhueta Simplificada pode ser obtida alterando a definição de silhueta de cada objeto, substituindo a Equação A.2 pela seguinte (HRUSCHKA; CAMPELLO; CASTRO, 2006):

$$s_{x_j} = \frac{b_{p,j}}{a_{p,j} + \varepsilon} \tag{A.4}$$

em que, ε é uma constante de baixo valor (e.g. 10^{-6} para dados normalizados) usada para evitar divisões por zero quando $a_{p,j} = 0$. Vale lembrar, assim como (VENDRAMIN; CAMPELLO; HRUSCHKA, 2010) pontua, que o intuito da Equação A.4 é o mesmo da Equação A.2, no sentido de que ambas visam favorecer valores altos para $b_{p,j}$ e valores baixos para $a_{p,j}$. A diferença entre as duas está em como o favorecimento é calculado, sendo que a Equação A.2 faz isso de forma linear e a Equação A.4 de forma não linear.

A.2.3 Calinski–Harabasz (VRC)

O critério de proporção de variância (do inglês, *Variance Ratio Criterion* (VRC)) (CALIńSKI; HARABASZ, 1974) avalia a qualidade de um agrupamento da seguinte forma:

$$VRC = \frac{traço(B)}{traço(W)} \times \frac{n-k}{k-1} \tag{A.5}$$

onde, W e B são matrizes de dispersão intra-*cluster* e entre-*cluster*, respectivamente — ambas de tamanho $a \times a$, lembrando que a é o número de atributos que descreve cada um dos objetos. Essas matrizes são definas por:

$$W = \sum_{l=1}^{k} W_l \tag{A.6}$$

$$W_l = \sum_{x_i \in C_l} (x_i - \bar{x}_l)(x_i - \bar{x}_l)^T \tag{A.7}$$

$$B = \sum_{l=1}^{k} n_l (\bar{x}_l - \bar{x})(\bar{x}_l - \bar{x})^T \tag{A.8}$$

onde, n_l é o número de objetos atribuídos ao *cluster* l (C_l), $\bar{x}_l$ é o vetor da média das amostras contidas naquele *cluster* (*centroid* do *cluster*) e $\bar{x}$ é o vetor da média global das amostras — isto é, o *centroid* dos dados ou média geral dos dados). Assim como pontua Vendramin, Campello e Hruschka (2010), o traço da matriz[1] de dispersão de intra-*cluster* (W) é a soma das variâncias intra-*cluster*. De forma similar, o traço da matriz B é a soma

[1] O traço é uma função matricial que resulta na soma dos elementos da diagonal principal de uma matriz.

das variâncias entre-*cluster*. Por isso, *clusters* compactos e bem separados tendem a ter valores baixos para o traço de W e apresentar valores altos para o traço de B. Logo, assim como lembra Naldi (2011), quanto mais os *clusters* da partição em questão apresentarem essas características maior será a razão entre o traço de W e o traço de B. Por fim, o termo de normalização $(n-k)/(k-1)$ evita que a razão entre os traços cresça monotonicamente, fazendo com que o VRC seja um índice de otimização em relação a k.

A.2.4 PBM

O critério PBM (PAKHIRA; BANDYOPADHYAY; MAULIK, 2004), também é baseado em distância intra-*cluster* e entre-*clusters*. Ele é dado pela Equação A.9, em que E_1 (Equação A.10) é uma constante que representa a soma das distâncias entre os objetos e a média geral dos dados, E_K (Equação A.11) representa a soma das distâncias intra-*clusters* e D_K é a distância máxima dentre todos os centroides (Equação A.12). De acordo essas equações, a melhor partição é indicada quando o valor do PBM é maximizado, isto é, o valor de D_K deve ser maximizado enquanto E_K deve ser minimizado.

$$PBM = \left(\frac{1}{k}\frac{E_1}{E_K}D_K\right)^2 \tag{A.9}$$

$$E_1 = \sum_{i=1}^{n} ||x_i - \bar{x}|| \tag{A.10}$$

$$E_K = \sum_{l=1}^{k} \sum_{x_i \in C_l} ||x_i - \bar{x}_l|| \tag{A.11}$$

$$D_K = \max_{l,m=1,\ldots,k} ||\bar{x}_l - \bar{x}_m|| \tag{A.12}$$

A.2.5 Davies–Bouldin (DB)

O índice Davies-Bouldin (DB) (DAVIES; BOULDIN, 1979) se dá pelo cálculo da razão entre a dispersão intra-*cluster* e a dispersão entre-*cluster*, por isso, é um índice relacionado ao VRC (NALDI, 2011). Esse índice avalia a qualidade um agrupamento de acordo com a Equação A.13, onde o termo D_l é dado pela Equação A.14. Na Equação A.14 o termo $D_{l,m}$ é espalhamento das distâncias intra-para-entre do l-ésimo e do m-ésimo *cluster*, dada pela Equação A.15. Na Equação A.15 os termos $\bar{d}_l$ e $\bar{d}_l$ representam as distâncias intra-*cluster* médias para o l-ésimo e o m-ésimo *cluster*, respectivamente, e $d_{l,m}$ é a distância entre-*cluster* dos mesmos. Essas distâncias estão descritas nas Equações A.16 e A.17, onde $||\,.\,||$ é a distância Euclidiana.

O termo D_l, da Equação A.13, representa o pior caso de espalhamento das distâncias intra-para-entre *clusters*. Minimizar D_l para todos os *clusters* acaba por minimizar o índice

DB. Logo, partições compactas e bem separadas são identificadas por valores menores de DB.

$$DB = \frac{1}{k}\sum_{l=1}^{k} D_l \tag{A.13}$$

$$D_l = max_{l \neq m}\{D_l, m\} \tag{A.14}$$

$$D_{l,m} = (\bar{d}_l + \bar{d}_m)/d_{l,m} \tag{A.15}$$

$$\bar{d}_l = \frac{1}{n_l}\sum_{x_i \in C_l} ||x_i - \bar{x}_l|| \tag{A.16}$$

$$d_{l,m} = ||\bar{x}_l - \bar{x}_m|| \tag{A.17}$$

A.2.6 Família de Índices Dunn

A família de índices Dunn é representada pela Equação A.18, em que δ_{C_p,C_q} é uma função de distância entres os *cluster* C_p e C_q; e Δ_{C_l} é o diâmetro do *cluster* C_l, que mede a dispersão do *cluster* (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2002). No índice Dunn original (DUNN, 1974), δ_{C_p,C_q} é dado pela Equação A.19 e Δ_{C_l} é dada pela Equação A.20. Contudo, em (BEZDEK; PAL, 1998), as definições de distância entre *clusters* e de diâmetro foram generalizadas, criando assim, 17 variantes do Dunn Index. Dentre elas a versão que é utilizada nesse trabalho, que é a mesma empregue por (NALDI; CARVALHO; CAMPELLO, 2013), onde δ_{C_p,C_q} passa a ser calculada utilizando a Equação A.21 e Δ_{C_l} definido pela Equação A.22.

$$DN = \underset{\substack{p,q \in \{1,\ldots,k\} \\ p \neq q}}{min} \left\{ \frac{\delta_{C_p,C_q}}{\underset{l \in \{1,\ldots,k\}}{max} \Delta_{C_l}} \right\} \tag{A.18}$$

$$\delta_{C_p,C_q} = \underset{\substack{x_i \in C_p, \\ x_j \in C_q}}{min} \quad d(x_i, xj) \tag{A.19}$$

$$\Delta_{C_l} = \underset{x_i, xj \in C_l}{max} \quad d(x_i, xj) \tag{A.20}$$

$$\delta_{C_p,C_q} = ||\bar{x}_p - \bar{x}_q|| \tag{A.21}$$

$$\Delta_{C_l} = \frac{2}{n_l}\sum_{x_i \in C_l} ||x_i - \bar{x}_l|| \tag{A.22}$$

Vale notar, assim como feito em (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2002; FACELI et al., 2011; NALDI; CARVALHO; CAMPELLO, 2013), que como as definições de δ_{C_p,C_q} e Δ_{C_l} estão ligadas aos conceitos de distância entre-*clusters* — preferencialmente com valores altos — e intra-*cluster* — preferencialmente com valores baixos —, que partições compostas de *clusters* compactos e bem separados são distinguidos por valores altos de DN em A.18.

A.2.7 Variância Intra-*cluster*

A variância intra-*cluster* avalia a qualidade de uma partição em termos da compactação de seus *clusters* (FACELI et al., 2011). Seu calculo é dado por:

$$VAR = \sqrt{\frac{1}{n}\sum_{c_k \in \pi}\sum_{x_i \in c_k} d(x_i, \mu_k)} \tag{A.23}$$

onde μ_k é o centroide do *cluster* e $d(.,.)$ é a função escolhida para cálculo de distância entre os objetos — no escopo desse trabalho utilizamos a distância Euclidiana.

A.2.8 Conectividade

Ligada ao conceito de encadeamento, a conectividade visa refletir o grau com o qual objetos vizinhos — isto é, aqueles com menor distância, significando maior similaridade — são atribuídos a um mesmo *cluster* (FACELI et al., 2011). Ela é calculada da seguinte forma:

$$CON = \sum_{x_i \in X}\sum_{j=1}^{NN} f(x_i, nn_{ij}) \tag{A.24}$$

$$f(x_i, nn_{ij}) = \begin{cases} 1/j & \text{se } x_i \in c_k, nn_{jj} \notin c_k \\ 0 & \text{caso contrário} \end{cases} \tag{A.25}$$

onde nn_{ij} é o $j-$ésimo vizinho mais próximo ao objeto x_i e NN é o parâmetro que delimita o número de vizinhos mais próximos que contribuem para a conectividade. Segundo (FACELI et al., 2011), quanto menor o valor da conectividade, melhor a partição.

APÊNDICE B – Ambiente de Testes e Detalhes de Implementação

B.1 Ambiente de Testes

Todos os experimentos realizados nesse trabalho foram executados na *Amazon Web Services* (AWS) [1], mais especificamente, através do serviço *Amazon Elastic Compute Cloud* (Amazon EC2) [2]. Para as todas as execuções foi utilizada uma máquina virtual (modelo `c5.18xlarge`), configurada com as seguintes especificações:

Processador: 72 vCPUs [3] de processadores Intel Xeon Platinum do modelo 8124M de 3 GHz, com Intel Turbo Boost de até 3.5 GHz.

Memória principal: 144 GB de RAM.

Memória secundária: A memória secundária não foi utilizada, todas as execuções utilizaram o TMPFS (SNYDER, 1990). Isso foi feito para diminuir o tempo de execução, pois, no MOCLE e nas seleções, há diversas tarefas que necessitam ler/escrever arquivos.

Sistema Operacional: Amazon Linux 64 *bits*, versão 2017.09.

B.2 Detalhes de Implementação

Abaixo estão destacadas como estão implementados os principais algoritmos citados e bibliotecas utilizadas nessa dissertação:

MOCLE: A versão do MOCLE aqui executada foi gentilmente cedida pela Profa. Dra. Katti Faceli, a principal autora do *framework*. O MOCLE foi implementado em C++ e utiliza, principalmente, duas bibliotecas, a *Multiple Objective MetaHeuristics Library in C++* (MOMHLIB++) (JASZKIEWICZ, 2005) e o *Serial Graph Partitioning and Fill-reducing Matrix Ordering* (METIS) (KARYPIS; KUMAR, 1998). A primeira biblioteca contém a implementação do NSGA-II, que é o algoritmo multiobjetivo baseado em Pareto do MOCLE. A segunda biblioteca é utilizada pelo operador de *crossover* do MOCLE.

[1] Disponível em: <https://aws.amazon.com/>
[2] Disponível em: <https://aws.amazon.com/ec2/>
[3] *Virtual Central Processing Unit*

SRD, SR e BRP: Todas essas seleções estão implementadas em Python, porém nem todos os índices relativos utilizados por elas estão em Python. Cinco deles são implementações da biblioteca *Environment for Developing KDD-Applications Supported by Index-Structures* (ELKI) (SCHUBERT et al., 2015) em Java. São os cinco: SS, SSA, VRC, PBM e DB. O índice Dunn está em Python e é uma implementação própria.

ASA: O ASA é uma reimplementação em Python da sua versão original em C++, ele está disponível em: <https://github.com/gpedote/asa>. Essa versão utiliza as bibliotecas: *NumPy* (WALT; COLBERT; VAROQUAUX, 2011), *Pandas* (MCKINNEY et al., 2010) e *Scikit-learn* (PEDREGOSA et al., 2011).

Diversidade, CAS e FILTA: Todos estão em Python, e também utilizam as mesmas bibliotecas do ASA.

HSS: O HSS está implementado em Python e foi gentilmente cedido pela Vanessa Antunes, a principal autora da técnica.

Testes Estatísticos: *SciPy* (JONES et al., 2001) (biblioteca em Python) e *The Pairwise Multiple Comparison of Mean Ranks Package* (PMCMR) (POHLERT, 2014) (biblioteca em R).

Gráficos: Os gráficos aqui contidos foram gerados com o uso da biblioteca *Matplotlib* (HUNTER, 2007).