



Harold Dias de Mello Junior

**Otimização de Funções Não Convexas utilizando um
Algoritmo de Estimação de Distribuição baseado em
Cópulas Multivariadas**

Tese de Doutorado

Tese apresentada como requisito parcial para
obtenção do título de Doutor pelo Programa de
Pós-Graduação em Engenharia Elétrica da PUC-Rio.

Orientadora: Profa. Marley Maria Bernardes Rebuszi Vellasco
Co-orientador: Prof. André Vargas Abs da Cruz



Harold Dias de Mello Junior

**Otimização de Funções Não Convexas utilizando um
Algoritmo de Estimação de Distribuição baseado em
Cópulas Multivariadas**

Tese apresentada como requisito parcial para obtenção do título de Doutor pelo Programa de Pós-Graduação em Engenharia Elétrica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Profa. Marley Maria Bernardes Rebuszi Vellasco

Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. André Vargas Abs da Cruz

Co-orientador

Centro Universitário Estadual da Zona Oeste

Prof. Fernando José Von Zuben

Universidade Estadual de Campinas

Prof. Valmir Carneiro Barbosa

Universidade Federal do Rio de Janeiro

Prof. Eduardo Fraga Lima de Melo

Universidade Estadual do Rio de Janeiro

Prof. Cristiano Augusto Coelho Fernandes

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Luis Martí Orosa

Departamento de Engenharia Elétrica – PUC-Rio

Prof. José Eugênio Leal

Coordenador Setorial do Centro

Técnico Científico - PUC-Rio

Rio de Janeiro, 29 de agosto de 2014

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Harold Dias de Mello Junior

Bacharel em Engenharia Elétrica pela Universidade Federal do Pará. Mestre na área de Telecomunicações pelo Departamento de Engenharia Elétrica da Universidade Federal do Pará.

Ficha Catalográfica

Mello Junior, Harold Dias de

Otimização de funções não convexas utilizando um algoritmo de estimação de distribuição baseado em cópulas multivariadas / Harold Dias de Mello Junior ; orientador: Marley Maria Bernardes Rebuszi Vellasco; co-orientador: André Vargas Abs da Cruz. – 2014.

137 f. : il. (color.); 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2014.
Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Otimização numérica contínua. 3. Computação evolutiva. 4. Algoritmos de estimação de distribuição. 5. Cópulas. I. Vellasco, Marley Maria Bernardes Rebuszi. II. Cruz, André Vargas Abs da. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

À minha esposa Nayme

Agradecimentos

Ao meu bom Deus pela benção da vida.

À minha orientadora professora Marley pela confiança depositada em mim e pela incomensurável generosidade.

Ao meu orientador André pelo incentivo contínuo e por todas as ideias e sugestões de investigação em incontáveis horas de reunião.

Às minhas mães, Célia e Amélia pela torcida diária e amor incondicional.

A todos os inúmeros amigos feitos na PUC, em especial, Marco, Edwin e Adriano, sempre atenciosos e dispostos a ajudar.

À minha esposa e amiga Nayme, pelo carinho e compreensão.

Ao CNPq e à PUC-Rio cujo suporte financeiro foi fundamental para a realização desse projeto.

Resumo

Mello Junior, Harold Dias de; Vellasco, Marley Maria Bernardes Rebuszi; Abs da Cruz, André Vargas. **Otimização de Funções Não Convexas utilizando um Algoritmo de Estimação de Distribuição baseado em Cópulas Multivariadas**. Rio de Janeiro, 2014. 137p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Algoritmos de estimação de distribuição (EDAs – *Estimation of Distribution Algorithms*) são uma classe de algoritmos evolutivos capazes de extrair e utilizar conhecimento ao longo do processo de busca. O passo mais importante e um gargalo, que estabelece diferenciação entre esses algoritmos, é a estimação da distribuição de probabilidade conjunta das soluções mais promissoras determinadas pela função de avaliação. Recentemente, uma nova abordagem baseada em teoria das cópulas foi desenvolvida. Este trabalho apresenta um algoritmo de estimação baseado em cópulas para problemas de otimização numérica. Este modelo implementa um EDA através da expansão multivariada de cópulas (EDA-MEC – *Estimation of Distribution Algorithm based on Multivariate Extension of Copulas*) para estimar a distribuição de probabilidade da qual é gerada uma população de indivíduos. O EDA-MEC difere de outros EDAs baseados em cópulas em alguns aspectos: o parâmetro de cópula é estimado de forma dinâmica, através de medidas de dependência; utiliza uma variação da distribuição de probabilidade aprendida para gerar indivíduos que ajudam a evitar a convergência prematura; e utiliza uma heurística para reinicializar a população ao longo da evolução elitista como uma técnica adicional para tentar preservar a diversidade de soluções. Após um conjunto de testes de parâmetros, inclusive das distribuições marginais, este trabalho mostra que estas abordagens melhoram o desempenho global da otimização comparativamente a outros EDAs baseados em cópulas, com a perspectiva promissora de ser um algoritmo competitivo frente a outras heurísticas comprovadamente eficientes, tais como a Estratégia Evolutiva com Adaptação da Matriz de Covariância (CMA-ES - *Covariance Matrix Adaptation Evolution Strategy*).

Palavras-chave

Otimização numérica contínua; computação evolutiva; algoritmos de estimação de distribuição; cópulas.

Abstract

Mello Junior, Harold Dias de; Vellasco, Marley Maria Bernardes Rebuszi (advisor); Abs da Cruz, André Vargas (advisor). **Nonconvex Functions Optimization using an Estimation of Distribution Algorithm based on Multivariate Copulas**. Rio de Janeiro, 2014. 137p. PhD Thesis — Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Estimation of distribution algorithms constitute a class of evolutionary algorithms that can extract and use knowledge acquired throughout the search process. Its most important step that differs most among EDAs, and also a bottleneck, is the estimation of the joint probability distribution associated with the variables from the most promising solutions determined by the evaluation function. Recently, a new approach to EDAs has been developed that is based on copula theory. This work presents a copula-based estimation of distribution algorithm for numeric optimization problems. This model implements an estimation of distribution algorithm using a Multivariate Extension of Copulas (EDA-MEC) to estimate the probability distribution for generating a population of individuals. EDA-MEC differs from other copula-based EDAs in some aspects: the copula parameter is estimated dynamically, using dependency measures; it uses a variation of the learned probability distribution to generate individuals that help to avoid premature convergence; and it uses a heuristic to reinitialize the population throughout an elitist evolution as an additional technique to try to preserve the diversity of solutions. After a set of parametric tests, including marginal distributions, this work shows that these approaches improve the overall performance of the optimization compared to other copula-based EDAs and promises to be a competitive algorithm compared to other efficient heuristics, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

Keywords

Continuous numeric optimization; evolutionary computation; estimation of distribution algorithms; copulas.

Sumário

| | | |
|--------|---|----|
| 1 | Introdução | 11 |
| 1.1. | Motivação | 11 |
| 1.2. | Objetivos e contribuições | 14 |
| 1.3. | Organização | 15 |
| 2 | Fundamentos de computação evolutiva | 16 |
| 2.1. | Introdução | 16 |
| 2.1.1. | Conceitos básicos de otimização | 17 |
| 2.1.2. | Tipos de problemas | 19 |
| 2.1.3. | Abordagens de solução | 20 |
| 2.2. | Computação evolutiva | 22 |
| 2.2.1. | Algoritmos genéticos | 24 |
| 2.2.2. | Estratégias evolutivas | 28 |
| 2.2.3. | Algoritmos de estimação de distribuição | 31 |
| 3 | Algoritmos de estimação de distribuição baseados em cópulas | 36 |
| 3.1. | Introdução à teoria de cópulas | 36 |
| 3.1.1. | Teoremas e Definições | 36 |
| 3.1.2. | Cópula condicional | 39 |
| 3.2. | Medidas de dependência | 40 |
| 3.2.1. | Coeficiente de correlação linear | 41 |
| 3.2.2. | Medidas de concordância | 41 |
| 3.2.3. | Dependência nas caudas | 43 |
| 3.3. | Cópulas elípticas | 44 |
| 3.3.1. | Cópula Gaussiana | 46 |
| 3.3.2. | Cópula t-Student | 47 |
| 3.4. | Cópulas arquimedianas | 49 |
| 3.4.1. | Cópulas bivariadas | 50 |
| 3.4.2. | Cópulas multivariadas | 54 |
| 3.5. | Métodos de estimação de parâmetros | 58 |

| | |
|---|-----|
| 3.5.1. Estimação por máxima verossimilhança | 58 |
| 3.5.2. Método de inferência para as marginais | 59 |
| 3.5.3. Método semiparamétrico | 59 |
| 3.5.4. Métodos não paramétricos | 60 |
| 3.6. Algoritmos de estimação baseados em cópulas (CEDAs) | 61 |
| 3.6.1. Revisão dos CEDAs existentes na literatura | 62 |
| | |
| 4 CEDA aplicado em problemas de otimização numérica (EDA-MEC) | 65 |
| 4.1. Algoritmos de estimação de distribuição | 65 |
| 4.1.1. Algoritmo de estimação de distribuição genérico | 65 |
| 4.1.2. Algoritmos de estimação de distribuição baseado em cópulas (CEDAs) | 67 |
| 4.2. Modelo proposto | 69 |
| 4.2.1. Seleção ou construção da cópula | 69 |
| 4.2.2. Estimação das distribuições marginais | 70 |
| 4.2.3. Estimação do(s) parâmetro(s) da cópula | 71 |
| 4.2.4. Amostragem da cópula | 73 |
| 4.2.5. Técnicas de diversidade | 78 |
| 4.2.6. Detalhamento do algoritmo do EDA-MEC | 80 |
| | |
| 5 Estudos de casos | 83 |
| 5.1. Propriedades de funções utilizadas como <i>benchmarks</i> | 83 |
| 5.2. Funções <i>benchmarks</i> utilizadas em otimização numérica | 85 |
| 5.3. Metodologia geral dos testes | 89 |
| 5.4. Testes de parametrização | 91 |
| 5.4.1. Ensaio com algoritmos de amostragem na cópula de Clayton | 91 |
| 5.4.2. Ensaio com as distribuições marginais | 94 |
| 5.4.3. Ensaio com números diferentes de indivíduos na população | 97 |
| 5.4.4. Ensaio com número de rebeldes | 100 |
| 5.4.5. Análise geral dos ensaios | 102 |
| 5.5. Comparação com outros CEDAs e com o CMA-ES | 111 |
| 5.6. Escalabilidade do EDA-MEC | 116 |
| 5.7. Outros ensaios | 117 |

| | |
|----------------------------------|-----|
| 6 Conclusões e trabalhos futuros | 119 |
| Referências bibliográficas | 123 |

1 Introdução

A execução de qualquer tarefa exige do agente a tomada de decisões. Na prática, frequentemente as tarefas são múltiplas e simultâneas, competindo entre si pelos mesmos recursos escassos de processamento, exigindo, desse modo, abordagens eficientes e sistemáticas. A modelagem destes problemas com a formulação de soluções define uma das áreas mais importantes de pesquisa em matemática aplicada, a de otimização. Quando a busca pela solução ótima envolve alguma forma de aleatoriedade, como a utilização de modelos probabilísticos, a otimização é do tipo estocástica (FOUSKAKIS & DRAPER, 2002). Este capítulo introdutório fornece motivação para este tópico, além da apresentação dos objetivos e contribuições e da organização desta tese.

1.1. Motivação

Conceitos e ferramentas de otimização têm sido largamente aplicados em métodos quantitativos de apoio à decisão para resolver extensa classe de problemas das mais diversas áreas de conhecimento, notadamente em engenharia. Para muitos destes, como o de determinação de parâmetros de um sistema de controle proposto por BALAKRISHNAN & BOYD (1992), é indispensável utilizar algoritmos de otimização global, visando encontrar a melhor solução – em um espaço de busca com múltiplas soluções locais – através da identificação das variáveis de decisão, x , que maximizam ou minimizam (dependendo do problema) uma função objetivo, $f(x)$, fornecida pelo usuário e sujeita a restrições em seu domínio.

Na literatura, entre as classificações existentes de estratégias de otimização global, PINTÉR (2002) as subdivide em exatas (determinísticas) e heurísticas ou semi-heurísticas (estocásticas), apresentando limitações inerentes que as tornam adequadas para tipos específicos de problemas. Métodos determinísticos, em teoria, garantem encontrar a solução global, diferentemente dos métodos estocásticos que

sequer garantem convergência para a solução global em um número finito de iterações. Entretanto, à medida que o problema se torna complexo com: o aumento do número de variáveis (dimensionalidade), uma função objetivo altamente não linear e com muitos ótimos locais, ou se o problema é de classe não determinístico polinomial (do inglês *NP-hard*); o custo computacional se torna excessivo para algoritmos determinísticos e, mesmo que estes dispusessem de recursos ilimitados de processamento, a intratabilidade desses problemas é irrefutável. Além disso, nesses casos, não há garantias matemáticas de que um determinado ponto seja ótimo global, pois os métodos numéricos clássicos podem estabelecer condições suficientes apenas para ótimos locais (GEORGIEVA & JORDANOV, 2009). Nestas circunstâncias, na ausência de propostas de solução determinísticas tratáveis computacionalmente, os algoritmos estocásticos tornam-se muito atrativos pelas grandes chances de identificar uma boa solução e até mesmo de alcançar o ótimo global, em um intervalo razoável de tempo.

Neste sentido, algoritmos evolutivos, tais como os algoritmos genéticos (HOLLAND, 1975; GOLDBERG, 1989; MICHALEWICZ, 1994), constituem meta-heurísticas populacionais de grande interesse de pesquisa devido aos resultados promissores obtidos em diversas aplicações (DAVIS, 1991; CHOU *et al.*, 2001; GRANELLI & MONTAGNA, 2007). Estas meta-heurísticas utilizam princípios da teoria de seleção natural de DARWIN (1859): a cada iteração do algoritmo ocorre uma seleção competitiva que escolhe as melhores soluções; estas, em seguida, são modificadas por operadores para gerar novas soluções, repetindo este ciclo, até que um dado critério de parada, definido pelo usuário, seja alcançado.

No mesmo grupo das meta-heurísticas populacionais, porém um pouco mais recentes que os algoritmos genéticos, estão os algoritmos de estimação de distribuição (da sigla em inglês EDA: *Estimation of Distribution Algorithm*). EDAs (MÜHLENBEIN & PAAß, 1996; HAUSCHILD & PELIKAN, 2011) são técnicas evolutivas que exploram regiões promissoras do espaço de busca através da amostragem de um modelo probabilístico estimado previa e periodicamente a partir de um conjunto de soluções candidatas selecionadas pelo algoritmo. Este mecanismo de aprendizagem e amostragem estatística, característico de métodos de aprendizado de máquina, permite que EDAs detectem explicitamente a relação de dependência entre as variáveis de decisão, transferindo-a para as novas soluções da

geração seguinte e, devido a isto, obtenham melhor desempenho que outros algoritmos evolutivos, uma vez que a otimização ocorre, em geral, com um número menor de avaliações da função objetivo (LARRAÑAGA & LOZANO, 2002).

A capacidade de resolver problemas grandes e complexos com desempenho superior ao de outras técnicas de otimização global (HAUSCHILD & PELIKAN, 2011) tem sido fator de grande motivação para o desenvolvimento de novas abordagens com esse tipo de algoritmo; todas, invariavelmente, buscando agregar métodos de estimação com melhor compromisso entre precisão e custo computacional de aprendizagem e amostragem do modelo probabilístico. Estes métodos têm conseguido construir distribuições probabilísticas mais representativas do espaço de buscas, em detrimento das distribuições gaussianas frequentemente utilizadas como aproximação do mesmo.

Alternativamente, a modelagem de EDAs pode ser feita através de funções cópulas. Cópulas, de acordo com o teorema de SKLAR (1959), são utilizadas para construir uma distribuição de probabilidade conjunta, relacionando a estrutura de dependência (estabelecida pela cópula) com as suas distribuições marginais univariadas. Esta técnica reduz o esforço computacional de aprendizagem do modelo e confere maior flexibilidade na modelagem de dados multivariados, gerando importantes contribuições em diversas aplicações, como na gestão de riscos financeiros (CHERUBINI *et al.*, 2004). No contexto da computação evolutiva, a utilização de cópulas em EDAs é ainda mais recente, sendo, portanto, tema de grande interesse de pesquisa. EDAs baseados em cópulas (SALINAS-GUTIÉRREZ *et al.*, 2009; WANG *et al.*, 2009a; WANG & ZENG, 2010; CUESTA-INFANTE *et al.*, 2010), a cada geração, produzem novas soluções de uma cópula – construída ou selecionada pelo algoritmo – das distribuições marginais univariadas. Apesar dos bons resultados obtidos em testes com algumas funções *benchmarks* de otimização numérica, aspectos da modelagem por cópulas em EDAs precisam ser investigados, incluindo as limitações das construções de cópulas multivariadas, a eficiência na relação entre o tipo de cópula e as distribuições marginais e mecanismos para manter a diversidade das soluções, para torná-la uma ferramenta competitiva em problemas de otimização global.

1.2. Objetivos e contribuições

Com base no exposto anteriormente, esta tese tem como objetivo principal a proposição e o desenvolvimento de um novo EDA baseado em cópulas aplicado em problemas de otimização numérica com as seguintes características:

- *Flexibilidade na construção das distribuições multivariadas* – A grande dificuldade de construção de modelos multivariados é resolvida através da teoria de cópulas. A abordagem utilizada permite criar modelos paramétricos e tratáveis de qualquer dimensão através da expansão multivariada de cópulas elípticas e arquimedianas;
- *Construção dinâmica de modelos probabilísticos* – A precisão do modelo probabilístico estimado por cópulas é função de um parâmetro relacionado à dependência estatística entre as variáveis. É desejável que este parâmetro retrate ao longo da evolução do algoritmo o grau correto de dependência;
- *Rápida Convergência* – O tempo de convergência depende da qualidade das soluções candidatas geradas e é medido em termos do número de chamadas da função objetivo ou de avaliação. Em decorrência da precisão do modelo construído pelas cópulas, do qual são geradas as novas soluções, um menor número de chamadas da função de avaliação ocorre;
- *Capacidade de escapar de ótimos locais* – Preservação da diversidade é o ponto chave de algoritmos evolutivos para evitar convergência prematura e estagnação em ótimos locais. Esta característica é fundamental para que algoritmos evolutivos obtenham sucesso na otimização de problemas mais complexos, tais quais os que envolvem multimodalidade;
- *Baixo custo computacional* – É bastante frequente em algoritmos de otimização com modelos estatísticos a realização de pelo menos uma estimação de parâmetros por máxima verossimilhança, o que constitui um procedimento de otimização a mais. Desse modo, é desejável que o algoritmo de otimização utilize métodos alternativos, menos custosos computacionalmente.

Em função dos objetivos definidos anteriormente, esta tese deixa como contribuição principal uma abordagem de EDAs baseados em cópulas mais eficiente que a de outros EDAs baseados em cópulas e promissora relativamente a outros algoritmos evolutivos. Esta abordagem considera:

- Construção de modelos probabilísticos com melhor relação entre precisão e esforço computacional;
- Estimação dos parâmetros das cópulas multivariadas através de medidas de estatísticas da população;
- Atualização dinâmica dos parâmetros das cópulas a cada geração do algoritmo, conferindo maior confiabilidade na estimação do modelo;
- Utilização de variações do modelo probabilístico estimado para inserir diversidade na população, evitando a convergência prematura;
- Utilização de uma heurística de reinicialização adaptativa e elitista da população como mecanismo de correção de convergência e, adicionalmente, de diversidade.

1.3. Organização

Este trabalho está estruturado da seguinte forma:

- O capítulo 2 apresenta os fundamentos de otimização e dos mecanismos e características dos principais algoritmos evolutivos, incluindo os EDAs;
- O capítulo 3 apresenta conceitos de dependência estatística entre variáveis, aspectos mais importantes da teoria de cópulas, e realiza uma breve revisão dos EDAs baseados em cópulas;
- O capítulo 4 apresenta os detalhes do modelo proposto nesta tese, o EDA-MEC (*Estimation of Distribution Algorithm based on Multivariate Extension of Copulas* – Algoritmo de Estimação de Distribuição baseado em Expansão Multivariada de Cópulas);
- O capítulo 5 mostra os resultados obtidos em diversos estudos de casos;
- O capítulo 6 apresenta discussões sobre o modelo e os resultados encontrados, e aponta direções para trabalhos futuros.

2 Fundamentos de computação evolutiva

2.1. Introdução

O conceito de otimização está associado à tomada de decisões que melhorem o desempenho e/ou custos de um processo ou fenômeno descrito por um modelo físico-matemático denominado de função objetivo. Esta formulação, assim como todas as outras de métodos de análise quantitativa, consiste em apenas uma aproximação do mundo real, e deve atender a requisitos conflitantes entre precisão (representação de todos os aspectos importantes do problema) e facilidade de tratamento computacional (viabilidade de se obter uma solução).

Segundo LUENBERGER & YINYU (2008), um problema de otimização envolve a seleção de valores para as variáveis inter-relacionadas de uma função objetivo, sendo os mesmos – e, portanto, a qualidade da decisão – avaliados pelo desempenho numérico alcançado em um ou mais objetivos de interesse. Muitas das vezes as tarefas são complexas: muitas variáveis, descritas por funções não lineares e, frequentemente, com múltiplas soluções ótimas locais. Na prática, o número dessas soluções é desconhecido e a qualidade delas pode diferir significativamente para a da solução global. Neste caso, conceitos e técnicas de otimização global precisam ser utilizados para encontrar o melhor conjunto de variáveis, responsável por maximizar ou minimizar, dependendo do contexto, a função objetivo, que pode ainda estar sujeita a restrições na escolha das variáveis.

Para que a relevância da otimização global, em crescente número de aplicações, possa ser percebida mais claramente, este capítulo aborda conceitos básicos de otimização, tipos de problemas e soluções, com ênfase nas características de três algoritmos da computação evolutiva: algoritmos genéticos, estratégias evolutivas e algoritmos de estimação de distribuição. A escolha destes é justificada dessa forma: algoritmos de estimação de distribuição, base do modelo apresentado nesta tese, foram criados para corrigir deficiências do método evolutivo mais conhecido, os algoritmos genéticos; e estratégias evolutivas possuem um dos

algoritmos mais eficientes em problemas de otimização numérica, o CMA-ES (HANSEN & OSTEMEIER, 2001; HANSEN *et al.*, 2011).

2.1.1. Conceitos básicos de otimização

Os seguintes elementos estão presentes em qualquer tarefa de otimização:

- Variáveis de decisão: são os elementos do modelo controlados pelo tomador de decisão e cujos valores determinam a solução do mesmo. As variáveis podem ser contínuas ou discretas.
- Função objetivo: modelo matemático fornecido pelo usuário e que indica o que se quer ver presente na solução. Sobre o modelo atuará um algoritmo de otimização, buscando identificar as soluções que maximizam ou minimizam a referida função quanto a um (simples) ou mais critérios (multiobjetivo). A literatura utiliza o termo caixa-preta para especificar um tipo de sistema de otimização em que o algoritmo desconhece completamente o comportamento da função objetivo.
- Espaço de busca ou domínio: compreende as possíveis soluções do problema (soluções candidatas), podendo ser delimitado por funções de restrição que especificam limitações na escolha das variáveis de decisão, tornando o processo de otimização ainda mais complexo. Se existirem tais restrições, o problema é chamado de otimização condicionada e passam a existir soluções candidatas factíveis e infactíveis.
- Solução ótima: conjunto de variáveis que maximiza ou minimiza globalmente uma função objetivo.

Representando matematicamente os conceitos supracitados, um problema de otimização, segundo a formalização de VANDERPLAATS (1984) consiste em:

$$\text{Minimizar: } F(\mathbf{X}) \qquad \text{função objetivo} \qquad (2-1)$$

sujeito a:

$$g_j(\mathbf{X}) \geq 0 \qquad j = 1, \dots, m \quad \text{restrições de desigualdade} \quad (2-2)$$

$$h_k(\mathbf{X}) = 0 \qquad k = 1, \dots, l \quad \text{restrições de igualdade} \quad (2-3)$$

$$X_i^{\min} \leq X_i \leq X_i^{\max} \qquad i = 1, \dots, n \quad \text{espaço de busca} \quad (2-4)$$

$$\text{onde } \mathbf{X} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{Bmatrix} \quad \text{variáveis de decisão}$$

Algumas informações pertinentes acerca das equações anteriores:

- i) O fato de formular o problema como uma minimização, em vez de maximização, ou utilizar $g_j(\mathbf{X}) \geq 0$ e não $g_j(\mathbf{X}) \leq 0$, é mera convenção, não alterando a essência dos conceitos apresentados;
- ii) A função objetivo, dada pela equação (2-1), e as funções de restrições definidas pelas equações (2-2) e (2-3) podem ser lineares ou não lineares em \mathbf{X} ;
- iii) Muito embora a equação (2-4) possa ser incluída no conjunto de restrições da equação (2-2), é usual a mesma ser expressa separadamente para delimitar de modo inequívoco a região de busca da solução ótima.

Em um dado problema, podem existir várias soluções ótimas locais, sendo a melhor delas de ótimo global. Formalmente, essa diferença entre mínimos locais e global utiliza o conceito de vizinhança: a partir de um ponto x qualquer de um espaço de busca B , define-se uma função de vizinhança, $V(x)$, como um subespaço de B , formado por uma região em torno de x . Assim, para um problema de otimização F , um ponto x será considerado ótimo local com respeito a uma vizinhança $V(x)$ se:

$$F(x) \leq F(y), \quad \forall y \in V(x) \quad (2-5)$$

E será considerado ótimo global se:

$$F(x) \leq F(y), \quad \forall y \in B \quad (2-6)$$

Em outras palavras, se $V(x)$ contiver todas as soluções candidatas do problema, x será ótimo global.

Um dos grandes desafios dos algoritmos de otimização é a necessidade de restringir o tamanho da vizinhança com a finalidade de reduzir o esforço computacional, sem, no entanto, comprometer a precisão do método, avaliando corretamente ótimos locais como soluções intermediárias em problemas de otimização global.

2.1.2. Tipos de problemas

Problemas de otimização usualmente são classificados em árvore, na qual a primeira divisão considera o tipo de domínio. A árvore a seguir, adaptada de ZABINSKY (2003) e BIEGLER (2010), divide-os, neste nível, em otimização contínua e discreta puras e em uma categoria de variáveis contínuas e discretas conhecida como problemas de otimização inteira mista.

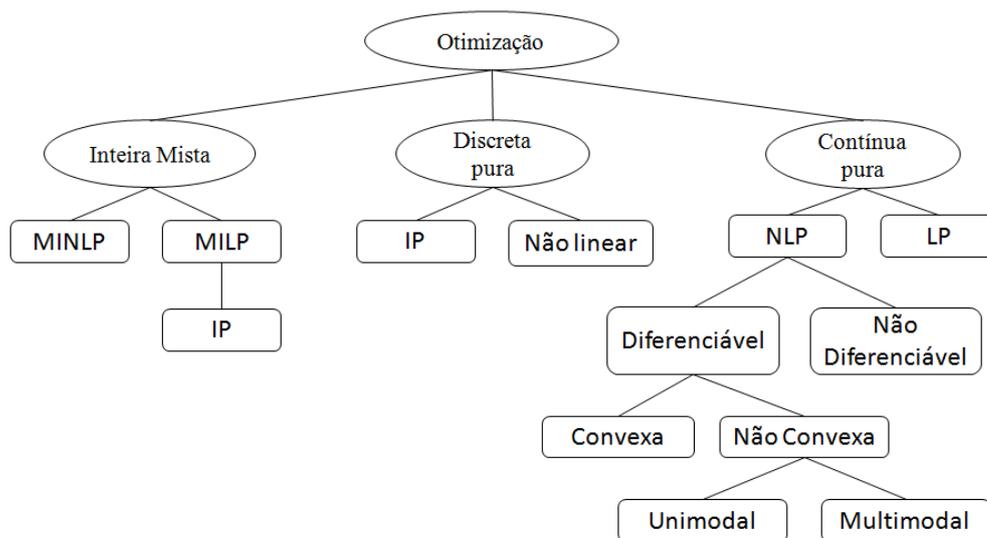


Figura 2.1 - Classificação dos problemas de otimização.

Seguindo a ramificação dos problemas de otimização inteira mista, os mais gerais destes são os de programação não linear inteira mista (MINLP – *Mixed Integer Nonlinear Programming*), nos quais a função objetivo ou uma das funções de restrições é não linear. Por outro lado, se ambas as referidas funções forem lineares, então os problemas serão de programação linear inteira mista (MILP – *Mixed Integer Linear Programming*). Um caso particular desta, quando todas as variáveis são inteiras, é denominado de problema de programação inteira (IP – *Integer Programming*) ou combinatória. Um exemplo clássico de otimização combinatória é o do caixeiro viajante (DANTZIG *et al.*, 1954) que consiste em determinar a menor rota a ser percorrida por um vendedor para visitar um número de n cidades, partindo de uma delas e retornando à mesma, ao final.

No caso dos problemas de otimização discreta ou contínua puras, representados no centro e no lado esquerdo do diagrama da figura 2.1, respectivamente, a linearidade da função objetivo e das funções de restrições

também é utilizada para determinar o nível seguinte da árvore. Problemas de otimização contínua pura serão de programação linear (LP) ou não linear (NLP), sendo estes últimos desdobrados, posteriormente, com base em duas outras características da função objetivo: diferenciabilidade e convexidade. A convexidade é expressa pela derivada de segunda ordem da função objetivo, $f(x)$, quando $f''(x) > 0$; caso $f''(x) < 0$, a função é do tipo côncava. Se um problema NLP é convexo, pode ser demonstrado que qualquer ótimo local também é um ótimo global, isto é, não haverá solução melhor para esse NLP. Por outro lado, problemas não convexos podem ter múltiplos ótimos locais, recaindo na categoria das funções multimodais.

Muito embora problemas de domínio discreto com funções objetivo não lineares também sejam de otimização global, esta tese aborda exclusivamente problemas de domínio contínuo, envolvendo a otimização numérica de funções não convexas, conforme estudos de casos apresentados no capítulo 5. É importante ressaltar que esta escolha não exclui a aplicabilidade do modelo proposto nesta tese em problemas de domínio contínuo com funções objetivo não diferenciáveis.

2.1.3. Abordagens de solução

A literatura classifica estratégias de otimização global de vários modos. NEUMAIER (2004) utiliza um critério muito específico baseado no grau de rigor com que uma abordagem alcança o resultado. Assim, o método pode ser:

- Incompleto: podem ficar presos em ótimos locais;
- Assintoticamente completo: alcançam o ótimo global com certeza se o tempo de busca for indefinidamente longo. Porém, não há como determinar o momento em que isto ocorre;
- Completo: alcançam o ótimo global, através de cálculos exatos e tempo de execução indeterminado. Encontram um valor aproximado do mesmo, com tolerâncias especificadas, em um tempo finito;
- Rigoroso: identificam o ótimo global com certeza, com tolerâncias especificadas, mesmo que haja erros de arredondamento.

Apesar de menos específica, a classificação mais comum é baseada no método de operação do algoritmo que os divide em duas categorias: determinísticos

e probabilísticos. O termo determinístico, frequentemente utilizado para caracterizar métodos completos e rigorosos, torna-se vago diante da classificação proposta por NEUMAIER (2004), pois, segundo este autor, muitos métodos incompletos e assintoticamente completos também são determinísticos.

Outra classificação dos algoritmos de otimização os define, quanto à metodologia de busca de soluções, em métodos exatos e heurísticos. Os primeiros, em geral determinísticos, baseados em informações do gradiente e da matriz hessiana, garantem, a rigor, encontrar um ótimo local, porém com um custo computacional que facilmente se torna inviável para modelos complexos, de grande escala e com não linearidades fortes. Considerando que a maioria dos problemas de otimização global contínua e discreta são do tipo *NP-hard* (não determinístico polinomial), caracterizada pela ausência de uma solução exata que possa ser obtida em tempo polinomial, métodos alternativos aos exatos precisam ser empregados. Nestes problemas, as heurísticas e meta-heurísticas, que possuem um componente global de busca estocástica, podem torná-los tratáveis.

Métodos heurísticos são aqueles que utilizam alguma informação do problema ou de um conjunto de soluções do mesmo com a finalidade de orientar o processo de busca de uma solução de alta qualidade, não necessariamente o ótimo global (MICHALEWICZ & FOGEL, 2000). Nos últimos anos, problemas complexos, tais como os *NP-hard*, motivaram o desenvolvimento de novas heurísticas, mais generalizadas, denominadas de meta-heurísticas, obtidas, em geral, pela combinação de heurísticas, e capazes de resolver diversos tipos de problemas sem a necessidade de grandes alterações no algoritmo de otimização. Um tipo particular, denominado de meta-heurísticas populacionais – que utiliza um conjunto (população) com mais de uma solução a cada etapa do processo de otimização – tem sido frequentemente aplicada. Embora não haja garantias matemáticas de convergência para o ótimo global, métodos meta-heurísticos têm sido bem sucedidos em diversas aplicações, encontrando soluções satisfatórias a custo computacional exequível (GOLDBERG, 1989; HARIK *et al.*, 2006; BIANCHI *et al.*, 2009; TALBI, 2009; SUN *et al.*, 2012). Outra vantagem desses métodos reside na simplicidade relativa de implementação: como, tipicamente, dependem apenas de avaliações da função objetivo, sem realizar cálculos diferenciais, podem ser codificados rapidamente e aplicados a uma extensa classe de problemas de otimização global, incluindo os do tipo mal definidos ou não

estruturados. Entretanto, possuem a desvantagem de requerer ajustes de parâmetros para cada problema, normalmente através de tentativas e erros.

Uma classe de meta-heurísticas de grande interesse para a área de otimização global é a dos algoritmos de computação evolutiva, cujo funcionamento é inspirado no comportamento evolutivo de sistemas biológicos e que serão discutidos na seção seguinte.

2.2. Computação evolutiva

A computação evolutiva é uma subárea da inteligência artificial, mais particularmente da inteligência computacional (ZURADA *et al.*, 1994), que abrange várias técnicas inspiradas em mecanismos de evolução biológica e que são consideradas métodos estocásticos de otimização global. A maioria desses métodos utiliza meta-heurísticas populacionais e são agrupados em:

- Algoritmos evolutivos;
- Algoritmos de inteligência coletiva (*swarm computing*);

A computação evolutiva inclui ainda muitas outras técnicas, algumas destas com pouca ou nenhuma inspiração biológica:

- Algoritmos meméticos (MOSCATO, 2002);
- Busca harmônica (GEEM *et al.*, 2001)

Além desses, existem muitos sistemas híbridos que agregam várias características dos algoritmos supracitados e que, em geral, não estão presentes em taxonomias da área, sendo referenciados simplesmente como métodos de computação evolutiva.

Segundo VON ZUBEN (2000), a vantagem mais significativa da computação evolutiva está na robustez e flexibilidade dos seus algoritmos, baseados em procedimentos genéricos e adaptáveis, que permitem resolver problemas complexos, nos quais métodos tradicionais são ineficazes. Baseando-se na herança genética e na teoria da sobrevivência das espécies (DARWIN, 1859), os algoritmos evolutivos compartilham elementos em comum:

- Uma representação genética (codificação) para a população de indivíduos, que representam soluções potenciais ou candidatas de um problema de otimização. Uma codificação especificada para um

problema, que pode não ser única, caracteriza o espaço de busca. Isto é, a codificação está diretamente vinculada à definição do espaço de busca;

- Uma função objetivo ou de avaliação para pontuar soluções em termos de aptidão;
- Operadores genéticos que permitam criar novos indivíduos, a partir de indivíduos já existentes, e inserir variabilidade nos descendentes;
- Um método de busca capaz de realizar a escolha dos melhores indivíduos de forma vinculada às suas respectivas aptidões (princípio da seleção natural).

Algoritmos evolutivos inicializam sua operação com a geração de uma população de soluções candidatas. Em seguida, após um mecanismo de seleção baseado na aptidão, soluções “pais” são escolhidas para gerar soluções “descendentes”. Estas surgem após a etapa de reprodução, momento em que operadores genéticos são aplicados nas soluções “pais”. Finalmente, as soluções resultantes substituem parcial ou totalmente a população original e um novo ciclo da evolução se inicia até que um ou mais critérios de parada, definidos previamente, sejam atendidos.

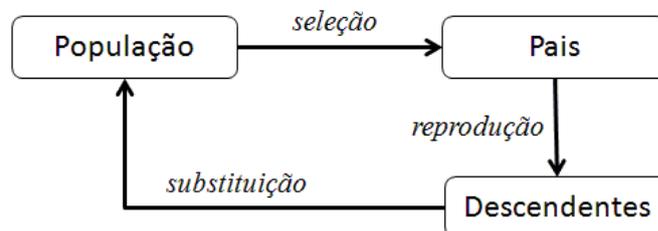


Figura 2.2 - Fluxograma de um algoritmo evolutivo. Adaptado de ABRAHAM *et al.* (2006)

A literatura relaciona cinco membros componentes da família de algoritmos evolutivos:

- Algoritmos genéticos [GA – *Genetic Algorithms*] (HOLLAND, 1975; GOLDBERG, 1989; MICHALEWICZ, 1994);
- Programação evolutiva [EP – *Evolutionary Programming*] (FOGEL *et al.*, 1966);
- Estratégias evolutivas [ES – *Evolution Strategies*] (RECHENBERG, 1973);
- Sistemas classificadores [CS – *Classifier Systems*] (BOOKER *et al.*, 1989);

- Programação genética [GP – *Genetic Programming*] (KOZA, 1992).

Constituindo uma variação dos algoritmos genéticos, os EDAs (MÜHLENBEIN & PAAß, 1996; HAUSCHILD & PELIKAN, 2011; CASTRO & VON ZUBEN, 2012) também são considerados algoritmos evolutivos, seguindo o mesmo fluxo de funcionamento dos demais, conforme representado na figura 2.2.

A diferença entre os algoritmos evolutivos está na forma como representam os dados, no tipo de operador genético aplicado e no modo de seleção dos indivíduos. Nas subseções seguintes, essas características serão destacadas em GAs, ES e EDAs.

2.2.1. Algoritmos genéticos

Sendo o mais conhecido dos algoritmos evolutivos, um algoritmo genético (GA – *Genetic Algorithm*) representa um indivíduo da população através de um cromossomo que contém a codificação (genótipo) de uma possível solução para o problema (fenótipo). Cromossomos são usualmente representados por listas de atributos ou vetores. Em sua versão clássica (HOLLAND, 1975), a codificação era binária de tamanho fixo. Posteriormente, MICHALEWICZ (1994) mostrou a superioridade da codificação em arranjos de números reais, no caso de problemas numéricos de alta dimensionalidade em que alta precisão é exigida, evidenciando que a codificação é uma das etapas críticas na definição de um algoritmo genético.

Outros aspectos, além da codificação, precisam ser especificados para definir completamente um GA. Um deles é o tamanho da população. Simulações indicam que esse parâmetro afeta diretamente o desempenho do algoritmo: quando muito pequena, dificulta a localização do ótimo global; por outro lado, se muito grande, torna o processo de busca ineficiente. De sorte que o tamanho adequado da população é normalmente obtido por experimentação.

Definido o tamanho da população e inicializando a mesma de forma aleatória, cada indivíduo desta é avaliado através da função objetivo do problema, produzindo uma medida de adaptação (*fitness*). O passo seguinte consiste em gerar novos indivíduos, a partir de um processo de modificação dos indivíduos mais adaptados, na etapa de reprodução. Esta envolve a aplicação dos operadores genéticos de

cruzamento ou recombinação (*crossover*) e mutação. A recombinação permuta importantes segmentos de diferentes indivíduos com uma probabilidade de ocorrência (p_r) por indivíduo, conhecida como taxa de recombinação, definida previamente pelo usuário. Por outro lado, a mutação é feita complementando o valor de posições do cromossomo, seguindo uma probabilidade (p_m) de mutação por posição, chamada de taxa de mutação e também parametrizada pelo usuário do algoritmo. Os GAs enfatizam a recombinação como o principal operador de busca, sendo a mutação um operador secundário, aplicado com baixas probabilidades, com o intuito de criar uma variabilidade extra (diversidade) na população, mas sem destruir o progresso já obtido com a recombinação.

Cuidados especiais devem ser tomados em problemas de otimização com restrições, pois a codificação adotada pode fazer com que os indivíduos modificados por esses operadores genéticos sejam inválidos. A figura 2.3 ilustra a aplicação desses operadores em cromossomos binários, concebidos para o primeiro tipo de algoritmo genético, o GA clássico. No item (a) desta figura, o operador de *crossover* de um ponto é aplicado em um ponto de corte “ k ” aleatoriamente escolhido na cadeia de bits dos indivíduos “pais”; os indivíduos “descendentes” são gerados com a troca dos segmentos da cadeia a partir de “ k ”. Outro tipo de *crossover* muito utilizado é o *crossover* uniforme (SYSWERDA, 1989): para cada bit no primeiro “descendente” é decidido (com alguma probabilidade fixa p_c) qual “pai” vai contribuir com seu valor para aquela posição. Como o *crossover* uniforme troca bits ao invés de segmentos de bits (que neste caso fazem o papel dos genes), ele pode combinar características independentemente da sua posição relativa no cromossomo. No entanto, não há nenhum operador de *crossover* que claramente apresente um desempenho superior aos demais. VON ZUBEN (2000) assevera que cada operador de *crossover* é particularmente eficiente para uma determinada classe de problemas e extremamente ineficiente para outras. No item (b) da figura 2.3, o operador de mutação atua sobre um indivíduo, alterando cada gene independentemente com uma probabilidade p_m , denominada de taxa de mutação. Usualmente, são atribuídos valores pequenos para esta taxa. A ideia intuitiva por trás do operador de mutação é criar uma variabilidade extra na população, mas sem destruir o progresso já obtido com a busca. Considerando codificação binária, o operador de mutação padrão simplesmente troca o valor de um gene em um

cromossomo. Assim, se um gene selecionado para mutação tem valor 1, o seu valor passará a ser 0 após a aplicação da mutação, e vice versa.

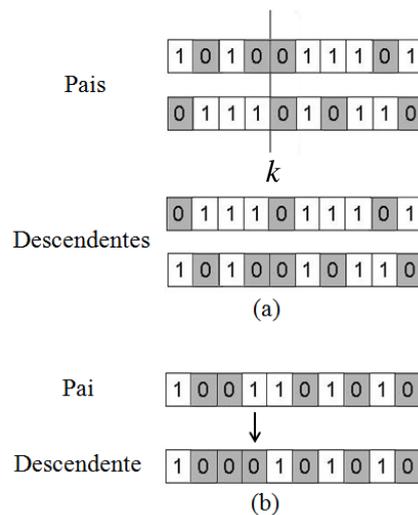


Figura 2.3 - Operadores genéticos: (a) *crossover* de um ponto (b) mutação pontual.

Das etapas gerais descritas na figura 2.2, o último passo do GA consiste na seleção dos indivíduos para a próxima geração, isto é, para a próxima iteração do algoritmo. O GA clássico faz uso do método da roleta, no qual cada indivíduo que compõe a população recebe uma probabilidade diretamente proporcional a sua aptidão relativamente ao somatório das aptidões dos demais, sendo estabelecida, dessa forma, a chance de um indivíduo passar para a próxima geração. Dada a natureza estocástica do mecanismo que poderia fazer com que o melhor indivíduo fosse perdido de uma geração para outra, outras estratégias foram elaboradas, tais como a seleção elitista, que sempre mantém o melhor indivíduo na geração seguinte, e a seleção baseada em *rank*, que faz uso da ordem das posições dos indivíduos, de acordo com suas aptidões, para determinar a probabilidade de seleção. Um dos métodos mais refinados é a seleção por torneio, por permitir ajustar a pressão seletiva, parâmetro que determina a dominância de um indivíduo na população e, por conseguinte, a perda de diversidade da população. Neste método, para se selecionar N indivíduos, realizam-se N torneios envolvendo q indivíduos em cada torneio, escolhidos sem levar em conta a aptidão e com reposição (um indivíduo pode aparecer mais de uma vez num mesmo torneio). Vence cada torneio aquele que apresentar a maior aptidão (comparado ao de seu(s) oponente(s) no torneio). De acordo com VON ZUBEN (2011), para propósitos práticos, $q > 10$ conduz a uma forte pressão seletiva, enquanto valores de q entre 3 e 5 levam a uma

fraca pressão seletiva, o que permite maior exploração do espaço de busca e, portanto, uma convergência mais lenta do algoritmo.

O ciclo da figura 2.2 se repete até que um número pré-determinado de iterações seja atingido ou até que um nível de adaptação esperado seja alcançado.

A codificação em arranjos de números reais, os mecanismos alternativos de seleção e o desenvolvimento de outros operadores de recombinação e mutação fizeram surgir, ao longo dos anos, diversas variações do GA clássico que quase sempre obtiveram desempenho superior a este em muitos casos, tais quais em problemas numéricos de alta dimensionalidade e em problemas de controle (MICHALEWICZ, 1994).

Um último aspecto de grande relevância neste tipo de algoritmo evolutivo diz respeito à modelagem teórica dos GAs. HOLLAND (1975) introduziu a noção de um *esquema* como um padrão geral de sequência de bits que é descrito por um molde constituído por 1, 0 e # (não importa). Por exemplo, o esquema 1 0 # # 0 1 representa a família de sequências de 6 bits, começando com 1 0 e terminando com 0 1. Como o padrão central # # descreve 0 0, 0 1, 1 0 e 1 1, o esquema completo representa quatro padrões de 1's e 0's. A fundamentação do GA clássico é que os esquemas são blocos construtivos de famílias de soluções. A teoria dos esquemas e a hipótese dos blocos construtivos (GOLDBERG, 1989), que figuram entre os conceitos mais aceitos para explicar o funcionamento do GA clássico, afirmam que o algoritmo manipula implicitamente um grande número de blocos construtivos de alta aptidão, através dos mecanismos de seleção e recombinação, reproduzindo e mesclando-os (PELIKAN *et al.*, 2000). Entretanto, em alguns tipos de problemas, a aleatoriedade dos operadores genéticos rompe os blocos construtivos, fazendo com que o algoritmo não consiga manipulá-los de maneira eficiente. Isto ocorre em cromossomos com alta epistasia, isto é, naqueles em que um gene influencia fortemente os outros. O problema de rompimento de blocos construtivos é muitas vezes referenciado como problema de ligação (*linkage problem*).

Desse modo, embora sejam verificadas aplicações bem sucedidas deste algoritmo, muitas vezes os operadores genéticos não contribuem para que os GAs alcancem o ótimo global. Este é o caso dos problemas de decepção (MICHALEWICZ, 1996), nos quais os blocos construtivos direcionam erroneamente o GA, fazendo-o convergir para subótimos.

2.2.2. Estratégias evolutivas

Estratégias evolutivas (ES – *Evolution Strategies*) são heurísticas de otimização baseadas em ideias de evolução e adaptação. Criada originalmente para dar tratamento a um problema de hidrodinâmica (RECHENBERG, 1973), esta classe especial de algoritmos evolutivos tem sido aplicada, principalmente, na otimização de funções contínuas.

Nas ES, a codificação é real e os cromossomos incluem, além dos parâmetros a otimizar (vetor de atributos X), os parâmetros da estratégia (σ , θ) que se referem àqueles que controlam o processo evolutivo da busca, parametrizando o modelo de densidade de probabilidade das mutações. Desse modo, uma população de n indivíduos pode ser representada por $P = (c_1, c_2, \dots, c_{n-1}, c_n)$, onde o i -ésimo cromossomo c_i é definido como $c_i = (X_i, \sigma_i, \theta_i)$, onde $X_i = (X_{i1}, X_{i2}, \dots, X_{in-1}, X_{in})$ é o vetor de atributos e $\sigma_i = (\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in-1}, \sigma_{in})$ e $\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{in-1}, \theta_{in})$ são os conjuntos de parâmetros da estratégia, em geral, desvios padrões e ângulos de rotação dos eixos principais das hiperelipses.

Observando-se a estrutura do genótipo, verifica-se que os parâmetros da estratégia também sofrem um processo de evolução, denominado de autoadaptação, que é fundamental para o sucesso das ES em tarefas de otimização. Também é importante ressaltar que, embora o indivíduo da população tenha dois ou mais vetores, somente o vetor de atributos X é submetido à avaliação da aptidão.

No que diz respeito aos operadores genéticos, a mutação e a seleção estão presentes em todas as ES, sendo a recombinação a menos utilizada. Nas ES, a mutação consiste em uma perturbação no vetor de atributos obtida através da adição de um vetor aleatório z_i com distribuição gaussiana de média zero e desvio padrão σ_i , isto é, $x_i^{t+1} = x_i^t + z_i$, onde $z_i = N(0, \sigma_i)$, exclusivamente no caso unidimensional. Com base no desvio padrão utilizado, definem-se três tipos de mutação:

1. Global:

A mutação utiliza um único desvio padrão para todos os atributos de X_i :

$$z_i = N(0, I\sigma), \text{ onde } I \text{ é a matriz identidade;}$$

2. Local:

A mutação utiliza desvios padrões individuais $\sigma_1, \dots, \sigma_n$ para cada atributo de X_i : $z_i = N_i(0, \sigma_i^{t+1})$, $i = 1, \dots, n$.

3. Correlacionada:

É utilizada uma matriz completa de covariância C_i (definida pelos vetores σ_i e θ_i) para cada atributo de X_i : $z_i = N(0, C_i(\sigma_i^{t+1}, \theta_i^{t+1}))$.

Em se tratando do operador de recombinação, o mesmo pode gerar um ou dois descendentes quando aplicado a dois ou mais pais. Este operador modifica não apenas o vetor de atributos, mas também os parâmetros da estratégia, existindo diversas formas de combinar as características dos pais para a geração do(s) descendente(s), similares as dos GAs. Uma forma muito comum em ES é a recombinação discreta que combina aleatoriamente as características dos pais. Especificamente, o *crossover* aritmético (MICHALEWICZ, 1996) foi desenvolvido para uso com codificação em ponto flutuante e consiste na combinação linear de dois cromossomos: sejam x_1 e x_2 dois indivíduos selecionados para *crossover*, então os dois descendentes resultantes serão $x'_1 = \alpha x_1 + (1 - \alpha)x_2$ e $x'_2 = (1 - \alpha)x_1 + \alpha x_2$, onde α é um número aleatório pertencente ao intervalo $[0, 1]$. Este operador é particularmente apropriado para problemas de otimização numérica com restrições, onde a região factível é convexa. Isto porque, se x_1 e x_2 pertencem à região factível, combinações convexas de x_1 e x_2 serão também factíveis. Assim, garante-se que o *crossover* não gera indivíduos inválidos para o problema em questão (VON ZUBEN, 2000).

Os primeiros algoritmos de estratégia evolutiva, (1+1)-ES, operavam com um único indivíduo, que gerava um único descendente, através de mutação com parâmetro σ constante. Esses dois indivíduos eram confrontados em uma competição por sobrevivência, na qual a seleção eliminava a solução mais pobre. O novo indivíduo só seria aceito como novo membro da população se tivesse melhor aptidão. Caso contrário, a solução original permanecia.

Diante da ineficiência das ES com parâmetro constante, RECHENBERG (1973) propôs a primeira técnica adaptativa de controle de parâmetros, conhecida como regra de sucesso do 1/5: se a razão entre o número de mutações com sucesso e o total for maior do que 1/5, deve-se aumentar o desvio padrão; caso seja menor, deve-se diminuí-lo. A razão intuitiva por trás da regra de sucesso do 1/5 é o aumento

da eficiência da busca: se bem sucedida, a busca continua com passos maiores; se não, o passo deve ser reduzido. Contudo, esta técnica pode levar à convergência prematura para algumas classes de funções. Isto motivou um aprimoramento do método através do aumento do número de pais. O conceito de população foi introduzido na estratégia $(\mu+1)$ -ES, na qual μ pais geram apenas um descendente através de recombinação e mutação, substituindo o pior pai da população.

SCHWEFEL (1981) estendeu a família das ESs para trabalhar com populações de pais e descendentes múltiplos, criando as ESs multimembros $(\mu+\lambda)$ -ES e (μ, λ) -ES. Nestas, o parâmetro de estratégia σ_i faz parte da informação genética do indivíduo, estando sujeito, portanto, à mutação e recombinação, e efetuando, assim, a adaptação *online* (autoadaptação) durante o processo evolutivo. Com isso, a atualização do vetor de parâmetros σ_i deixa de ser realizada pela regra de sucesso do 1/5, sendo esta a principal diferença dessas ES para as outras variantes. Mais especificamente:

- Na $(\mu+\lambda)$ -ES, μ pais produzem $\lambda \geq \mu$ descendentes e a população $\mu+\lambda$ é posteriormente reduzida a μ melhores indivíduos. O mecanismo de seleção opera no conjunto $\mu+\lambda$, eliminando da população os pais à medida que geram descendentes com melhor aptidão;
- Na (μ, λ) -ES, μ pais produzem λ descendentes, sendo que a nova população de μ indivíduos é formada apenas por indivíduos selecionados do conjunto de λ descendentes. Desse modo, o período de vida de cada indivíduo está restrito a uma geração.

Uma abordagem mais recente que constitui o atual estado da arte em problemas de otimização contínua é a Estratégia Evolutiva com Adaptação da Matriz de Covariância (CMA-ES), proposta por HANSEN & OSTERMEIER (2001). A excelência dessa ES está no modo direto e não aleatório com que a mesma atualiza os parâmetros estratégicos. As informações obtidas do *caminho da evolução* – passos seguidos em d iterações do algoritmo – norteiam a intensidade das mudanças na matriz de covariância: aumentam, quando variações na direção do espaço são bem sucedidas e diminuem gradativamente em outras direções. A acumulação de informação torna possível adaptar confiavelmente a matriz de covariância até mesmo quando o algoritmo utiliza pequenas populações. Outra característica do CMA-ES é a invariância deste a transformações no espaço de

busca, incluindo rotação, reflexão e translação. A verificação destas propriedades, altamente desejáveis em algoritmos de otimização, implicam em uma padronização de comportamento do algoritmo em uma classe de funções e, portanto, implicam na generalização dos resultados empíricos obtidos (HANSEN *et al.*, 2011). Ademais, todas as comparações experimentais confirmam a superioridade desta abordagem sobre as estratégias evolutivas puramente mutativas.

2.2.3. Algoritmos de estimação de distribuição

Algoritmos de estimação de distribuição (EDAs – *Estimation of Distribution Algorithms*) possuem muitos aspectos em comum com os algoritmos genéticos, conforme é possível verificar na figura 2.4. Entretanto, em EDAs, novos indivíduos são gerados diferentemente. No lugar dos operadores genéticos de recombinação e mutação, EDAs estimam a distribuição de probabilidade das soluções mais promissoras e, em seguida, fazem uma amostragem da mesma para gerar novas soluções. Este mecanismo confere aos EDAs a capacidade de extrair e utilizar conhecimento ao longo da execução, inexistente nos GAs.

Outra importante diferença existente é o modo de construir soluções. Enquanto GAs tentam combinar blocos construtivos implicitamente, EDAs buscam encontrar explicitamente a correlação ou relação de dependência entre as variáveis. Neste ponto, destaca-se a principal vantagem dos EDAs sobre os outros algoritmos evolutivos: a capacidade de tratar, de modo robusto (com pouca variação de desempenho a cada execução) e escalável, problemas que contenham interações importantes entre variáveis (LOZANO *et al.*, 2006; PELIKAN, 2005).

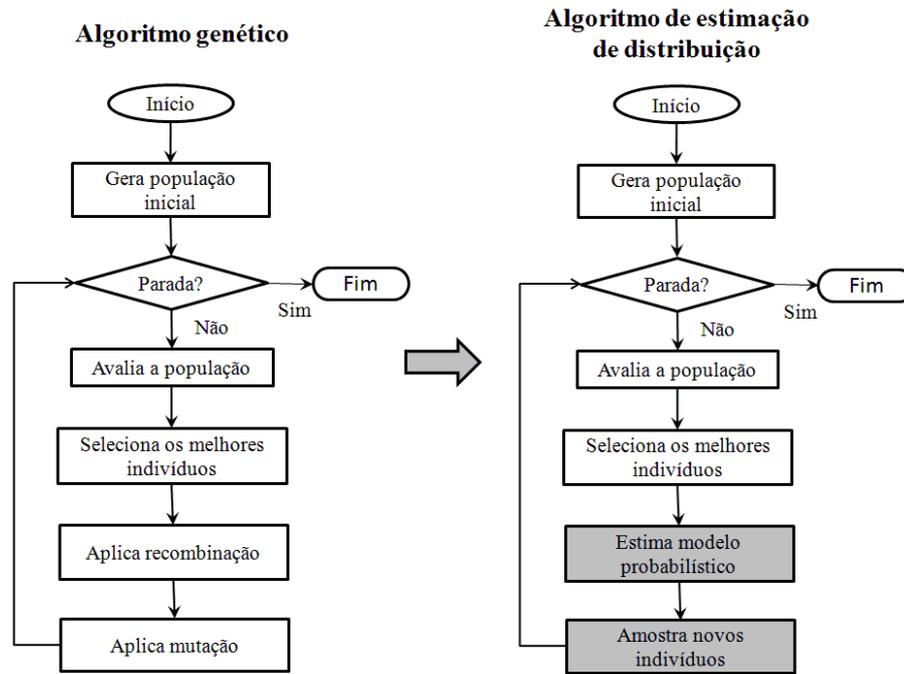


Figura 2.2 - Fluxograma de etapas de algoritmos genéticos e algoritmos de estimação de distribuição. Adaptada de VON ZUBEN (2011)

Assim, a etapa fundamental desta classe de algoritmos evolutivos é a estimação da distribuição de probabilidade conjunta das variáveis de entrada das soluções. Esta tarefa se torna mais difícil quando o problema em questão possui dependências complexas entre as variáveis, exigindo um compromisso entre precisão de estimação e custo computacional. Em função disto, a literatura usualmente classifica os EDAs quanto à complexidade do modelo utilizado: variáveis sem dependência; dependência aos pares; dependência multivariada e modelos com mistura de distribuições (LARRAÑAGA, 2001).

A estimação realizada pelos algoritmos com variáveis sem dependências é a mais simples. Supondo que não existe relação de dependência entre as variáveis do problema, a distribuição de probabilidade conjunta pode ser fatorada em n funções de densidade de probabilidade (fdp's) univariadas:

$$p(X) = \prod_{i=1}^n p(X_i) \quad (2-7)$$

onde X é o conjunto de soluções promissoras, em um espaço de busca n -dimensional. Essa abordagem tem baixo custo computacional, no entanto, falha em problemas mais difíceis. São EDAs univariados discretos: PBIL (*Population Based Incremental Learning*) (BALUJA, 1994), UMDA (*Univariate Marginal*

Distribution Algorithm) (MÜHLENBEIN & PAAß, 1996), cGA (*Compact Genetic Algorithm*) (HARIK *et al.*, 1999); e contínuo: UMDAc (LARRAÑAGA *et al.*, 1999).

Quando o algoritmo é de dependência aos pares, considera-se que as variáveis possuem dependência em relação a, no máximo, outra variável, sendo a distribuição de probabilidades fatorada como segue:

$$p(X) = \prod_{i=1}^n p(X_i | X_{j(i)}), \text{ com } i \neq j \quad (2-8)$$

onde $X_{j(i)}$ é a variável da qual X_i é dependente.

Enquanto os algoritmos de variáveis independentes estimam apenas os parâmetros do modelo, na dependência aos pares a aprendizagem paramétrica requer também a construção de uma rede de probabilidades (grafo) entre as variáveis para representar o modelo, o que acarreta em um aumento do custo computacional. São exemplos de algoritmos deste tipo: MIMIC (*Mutual Information Maximizing Input Clustering*) (DE BONET *et al.*, 1997), COMIT (*Combining Optimizers with Mutual Information Trees*) (BALUJA & DAVIES, 1997), BMDA (*Bivariate Marginal Distribution Algorithm*) (PELIKAN & MÜHLENBEIN, 1999a).

Apesar de modelos de dependência aos pares serem eficientes em certos cenários, em problemas multivariados os mesmos são insuficientes para representar adequadamente o espaço de soluções. Nestes casos, todas as dependências entre variáveis são modeladas por uma distribuição da seguinte forma:

$$p(X) = \prod_{i=1}^n p(X_i | X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_k) \quad (2-9)$$

Embora estes algoritmos demandem alto custo computacional, devido ao aprendizado de um número maior de relações de dependência entre variáveis, o número de avaliações da função objetivo tende a ser reduzido, uma vez que a amostragem dos novos indivíduos será realizada em um modelo mais preciso. São exemplos de EDAs de dependência multivariada: BOA (*Bayesian Optimization Algorithm*) (PELIKAN *et al.*, 1999b); MD-EDA (*Markov Network Estimation Distribution Algorithm*) (SANTANA, 2005); hBOA (*Hierarchical Bayesian Optimization Algorithm*) (PELIKAN, 2005); ECGA (*Extended Compact Genetic Algorithm*) (HARIK *et al.*, 2006).

Além dos modelos probabilísticos básicos, EDAs também têm utilizado outras técnicas de modelagem. Modelos com mistura de distribuições aumentam a flexibilidade da estimação da distribuição de probabilidade conjunta em EDAs, especialmente no caso de problemas de otimização multimodais. Estes modelos realizam agrupamento de soluções promissoras e posicionam distribuições de probabilidade junto a cada agrupamento. São descritos por:

$$p(X) = \sum_{i=1}^k \pi_i p_i(X) \quad (2-10)$$

onde π_i representa o peso da i -ésima componente da mistura, $p_i(x)$ é a distribuição de probabilidade do i -ésimo grupo criado pelo método de agrupamento de dados e k é o número de grupos. O peso de cada componente e a estrutura do modelo de mistura são obtidos através de algum procedimento dedicado, como, por exemplo, o algoritmo EM (*Expectation-Maximization*) (DEMPSTER et al., 1977). Para completar o modelo, é necessário informar o número de componentes de cada grupo, residindo, neste ponto, a principal dificuldade deste método. São EDAs do tipo modelo de mistura de distribuições: EMDA (*Estimation of Mixtures of Distributions Algorithm*) (PENA et al., 2001); mIDEA (*mixed Iterated Density Estimation Evolutionary Algorithm*) (BOSMAN & THIERENS, 2001); EDA_{OGMM} (*Online Gaussian Mixture Model Algorithm*) (GONÇALVES & VON ZUBEN, 2011). O mIDEA, em particular, constrói o modelo probabilístico através de uma mistura de distribuições normais que se ajustam a grupos de variáveis, utilizando a métrica do critério de informação bayesiano (BIC – *Bayesian information criterion*) (SCHWARZ, 1978) para avaliar as dependências entre as variáveis. Em seguida, combinando EDAs, seleção de truncamento baseada em dominância e agrupamento, THIERENS & BOSMAN (2001) propuseram extensões do mIDEA para problemas de otimização multiobjetivos, os algoritmos MIDEA (*Mixture-based Multiobjective Iterated Density-Estimation Evolutionary Algorithm*). Modificações adicionais permitiram validar a eficiência desses algoritmos em problemas mono-objetivo e multiobjetivos, conferindo premiações aos autores em edições da Conferência de Computação Genética e Evolutiva (GECCO – *Genetic and Evolutionary Computation Conference*): uso de técnicas para calcular e utilizar a informação do gradiente e de esquema adaptativo de alocação de recursos (BOSMAN & DE JONG, 2006); uso de memória para reduzir requisitos de

tamanho da população e divisão do espaço de busca (BOSMAN, 2009); uso de novo algoritmo de agrupamento para reduzir o risco de convergência prematura devido a superposição de soluções (BOSMAN, 2010).

Uma abordagem mais recente é a que utiliza a função de cópulas (SKLAR, 1959) para estimar o modelo probabilístico dos EDAs e que será discutida no próximo capítulo.

3

Algoritmos de estimação de distribuição baseados em cópulas

Este capítulo apresenta fundamentos de uma importante ferramenta em modelagem estatística e simulações, as funções cópulas. Na última seção, é realizado um levantamento da literatura, relacionando os principais trabalhos que utilizaram cópulas no contexto da computação evolutiva.

3.1. Introdução à teoria de cópulas

Introduzida por SKLAR (1959), a teoria de cópulas começou a ser aplicada extensivamente apenas na última década, na área de finanças. Reconhecida a impropriedade da distribuição normal multivariada na modelagem de dados financeiros, cópulas têm constituído ferramenta de grande utilidade por permitir a construção de modelos multivariados capazes de caracterizar adequadamente a estrutura de dependência das variáveis do problema.

Nas próximas seções, serão apresentados os fundamentos da teoria de cópulas, incluindo os princípios de construção e as técnicas de simulação.

3.1.1. Teoremas e Definições

Definição 3.1 (Cópula)

Seja $u = (u_1, \dots, u_n) \in [0,1]^n$ um vetor de n variáveis. Uma função $C(u): [0,1]^n \rightarrow [0,1]$ é uma cópula n -dimensional se, e somente se, satisfaz as seguintes propriedades básicas:

- $C(u) = u_k$, se todas as componentes de u forem iguais a 1, exceto u_k ;
- $C(u) = 0$, se ao menos uma componente de u for zero;
- C é crescente em cada componente u_k .

Um resultado importante, sempre citado, e base para aplicações, é o teorema de Sklar (SKLAR, 1959).

Teorema 3.1 (Sklar)

Segundo o teorema de Sklar, base da teoria das cópulas, a função de distribuição cumulativa conjunta (JCDF – *Joint Cumulative Distribution Function*), F , das variáveis aleatórias X_1, X_2, \dots, X_n com distribuições marginais F_1, F_2, \dots, F_n , respectivamente, pode ser caracterizada por uma cópula C n -dimensional, tal que para todo vetor $x \in \overline{\mathbb{R}^n}$:

$$\begin{aligned} F(x_1, x_2, \dots, x_n) &= C(F_1(x_1), F_2(x_2), \dots, F_n(x_n)) \\ &= P\{X_1 \leq x_1, X_2 \leq x_2, \dots, X_n \leq x_n\} \end{aligned} \quad (3-1)$$

Adicionalmente, o teorema de Sklar afirma que se F_1, F_2, \dots, F_n forem todas contínuas, então C será única. Neste caso, as distribuições marginais podem ser separadas da estrutura de dependência, sendo esta representada por uma cópula. Por outro lado, quando as distribuições marginais são discretas, não é possível assumir que a representação (3-1) seja única.

Como representação alternativa, a equação (3-1) pode ser escrita de forma invertida. Para qualquer vetor $u \in [0, 1]^n$:

$$C_F(u_1, u_2, \dots, u_n) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), \dots, F_n^{-1}(u_n)) \quad (3-2)$$

onde C_F denota a cópula pertinente a F e $F^{-1}(u) = \inf\{x \in \mathbb{R} \mid F_i(x_i) \geq u\}$, com $i = 1, \dots, n$ e $u \in (0, 1)$, constitui a inversa generalizada da função de distribuição marginal. Assim, uma cópula é uma distribuição multivariada com marginais uniformes no intervalo $(0, 1)$.

Definição 3.2 (Função conjunta e cópula de sobrevivência)

Seja \bar{F} a função conjunta de sobrevivência para n variáveis aleatórias X_1, X_2, \dots, X_n com função de distribuição conjunta F . Então, $\bar{F}(x_1, x_2, \dots, x_n) = P\{X_1 > x_1, X_2 > x_2, \dots, X_n > x_n\}$. Analogamente, \bar{C} representa a função conjunta de sobrevivência de U_i variáveis aleatórias uniformes em $(0, 1)$ e $0 \leq u_i \leq 1$, $i = 1, \dots, n$ e que tem a cópula C como função de distribuição, isto é:

$$\bar{C}(u_1, u_2, \dots, u_n) = P\{U_1 > u_1, U_2 > u_2, \dots, U_n > u_n\} \quad (3-3)$$

No caso bivariado, a cópula de sobrevivência $\hat{C}(u, v)$ associada à cópula $C(u, v)$ é definida como a cópula do vetor $(1 - U, 1 - V)$, onde (U, V) é um vetor de v.a's uniformemente distribuídas em $(0, 1)$ e que tem cópula $C(u, v)$. Então:

$$\hat{C}(u, v) = u + v - 1 + C(1 - u, 1 - v) \quad (3-4)$$

Teorema 3.2 (Independência entre variáveis)

De acordo com o teorema 2.4.2 de NELSEN (2006):

Sejam X_1, X_2, \dots, X_n variáveis aleatórias contínuas com cópula C e distribuições marginais F_1, F_2, \dots, F_n . Então, X_1, X_2, \dots, X_n são independentes se, e somente se,

$$C = F_1(X_1)F_2(X_2) \cdots F_n(X_n) = u_1u_2 \cdots u_n \quad (3-5)$$

A expressão anterior é denominada de cópula independente ou cópula produto. No caso bivariado, a figura seguinte ilustra perspectivas de um gráfico de superfície dentro do cubo unitário: nos eixos da base do primeiro quadrante estão as distribuições marginais e no eixo perpendicular, a cópula. Observe que a superfície é não decrescente, de acordo com o requisito da definição 3.1.

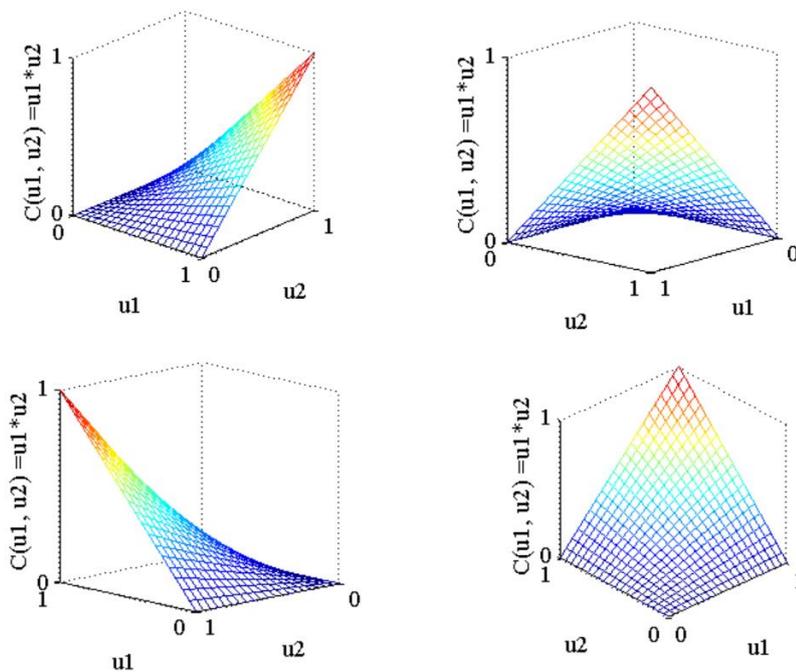


Figura 3.1 - Gráfico de superfície da cópula produto $C(u_1, u_2) = u_1u_2$ para $(u_1, u_2) \in [0, 1]^2$.

Teorema 3.3 (Invariância a transformações estritamente crescentes)

O teorema 2.4.3 de NELSEN (2006) afirma:

Seja C a cópula das variáveis aleatórias contínuas X_1, X_2, \dots, X_n e seja $RanX_i$, para $i = 1, 2, \dots, n$, o conjunto de valores que a variável X_i pode assumir. Se $\alpha_1, \alpha_2, \dots, \alpha_n$ forem funções contínuas e estritamente crescentes em $RanX_1, RanX_2, \dots, RanX_n$, respectivamente, então C também é a cópula de $\alpha_1(X_1), \alpha_2(X_2), \dots, \alpha_n(X_n)$.

Teorema 3.4 (Derivada parcial de cópulas)

Estendendo o teorema 2.2.7 de NELSEN (2006) para o caso n -variado:

Seja C uma cópula. Para qualquer $u_n \in [0,1]^n$, a derivada parcial $\frac{\partial}{\partial u_n} C(u_1, u_2, \dots, u_n)$ existe para quase todo u_n , e para tais u_1, u_2, \dots, u_n ,

$$0 \leq \frac{\partial}{\partial u_n} C(u_1, u_2, \dots, u_n) \leq 1 \quad (3-6)$$

Adicionalmente, as funções $u_n \mapsto \frac{\partial}{\partial u_n} C(u_1, u_2, \dots, u_n)$ são definidas e não decrescentes em quase toda parte em $[0,1]^n$.

Definição 3.3 (Função densidade de probabilidade conjunta)

A partir de (3-1) e do teorema anterior, é possível definir a função densidade de probabilidade conjunta (JPDF – *Joint Probability Density Function*), f , como a derivada de ordem n da cópula C em relação às suas funções de distribuição marginais, isto é:

$$\begin{aligned} f(x_1, \dots, x_n) &= \frac{\partial^n C(F_1(x_1), \dots, F_n(x_n))}{\partial F_1(x_1), \dots, \partial F_n(x_n)} \prod_{i=1}^n f_i(x_i) \\ &= c(F_1(x_1), \dots, F_n(x_n)) \prod_{i=1}^n f_i(x_i) \end{aligned} \quad (3-7)$$

onde F_i e f_i são as funções distribuição e densidade de probabilidades marginais, respectivamente, e

$$c(F_1(x_1), \dots, F_n(x_n)) = \frac{\partial^n C(u_1, \dots, u_n)}{\partial u_1 \dots \partial u_n} = \frac{f(x_1, \dots, x_n)}{f_1(x_1) \dots f_n(x_n)} \quad (3-8)$$

Na representação (3-7), conforme observa MENDES (2004), fica clara a decomposição da densidade conjunta $f(x_1, \dots, x_n)$ em duas partes: uma que descreve apenas a estrutura de dependência e outra que descreve o comportamento marginal de cada variável. Na prática, esta decomposição simplifica a especificação da distribuição multivariada e sua estimação (MENDES, 2009).

3.1.2. Cópula condicional

Através da função cópula é possível obter também a distribuição condicional. Se as variáveis aleatórias U_1, \dots, U_k possuem função de distribuição conjunta C , então a distribuição condicional de U_k dados os valores de U_1, \dots, U_{k-1} é dada por:

$$\begin{aligned}
C_k(u_k|u_1, \dots, u_{k-1}) &= P\{U_k \leq u_k | U_1 = u_1, \dots, U_{k-1} = u_{k-1}\} \\
&= \frac{\frac{\partial^{k-1} C_k(u_1, \dots, u_k)}{\partial u_1 \dots \partial u_{k-1}}}{\frac{\partial^{k-1} C_{k-1}(u_1, \dots, u_{k-1})}{\partial u_1 \dots \partial u_{k-1}}} \quad (3-9)
\end{aligned}$$

No caso bivariado e quando as marginais U_1 e U_2 são uniformes em $(0,1)$, (3-7) é simplificada a:

$$C_2(u_2|u_1) = \frac{\partial C_2(u_1, u_2)}{\partial u_2} \quad (3-10)$$

Esta última relação constitui a base de um algoritmo geral para geração de variáveis aleatórias de uma dada cópula, ineficiente na maioria dos casos e apresentado no capítulo seguinte. Entretanto, a relação (3-10) é também a base de um tipo robusto de construção de cópulas multivariadas, os modelos *pair-copulas* (AAS *et al.*, 2007).

3.2. Medidas de dependência

Esta seção apresenta alguns conceitos básicos sobre medidas de dependências entre variáveis aleatórias. Considerando que a condição de independência decorre, na maioria das vezes, de uma suposição, não verificada em muitos casos práticos, torna-se de suma importância o entendimento da relação entre elas para a correta modelagem estatística.

Uma vez que, de acordo com a seção anterior, as funções cópulas são capazes de fornecer informação sobre a estrutura de dependência, independentemente da especificação das distribuições marginais, e que, no caso contínuo, as referidas funções são invariantes a transformações estritamente crescentes das variáveis aleatórias, a teoria de cópulas está diretamente relacionada ao estudo de medidas de dependência entre variáveis aleatórias. Portanto, conceitos de dependência são propriedades de cópulas (NELSEN, 2006).

CHERUBINI *et al.* (2004) e outros autores têm utilizado um novo termo, distinto de dependência, denominado de medidas de associação, que se baseiam em relações de concordância. As próximas subseções apresentarão algumas definições relativas a quatro das medidas de dependência/associação mais utilizadas.

3.2.1. Coeficiente de correlação linear

O coeficiente de correlação linear ou de Pearson é definido para um par de variáveis aleatórias (X, Y) com variâncias finitas e diferentes de zero e é dado por:

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}} \quad (3-11)$$

onde $\text{Cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$ é a covariância de (X, Y) e $\text{Var}(X)$ e $\text{Var}(Y)$ são as variâncias de X e Y .

O coeficiente de correlação linear é uma medida de dependência frequentemente utilizada devido à simplicidade da sua estimação. Entretanto, este coeficiente tem as seguintes limitações:

- Mede somente correlação do tipo linear, capturando apenas o grau e o sinal da mesma;
- É uma medida de dependência natural das distribuições elípticas, em particular da distribuição normal multivariada, e, portanto, adequada apenas para estas (vide na seção 3.3 a definição de distribuição elíptica);
- Uma única observação pode ter uma grande influência no valor de ρ , isto é, ρ não é uma medida de dependência robusta. Devido a isto, o coeficiente de correlação linear não deve ser utilizado quando a amostra possui *outliers*.

Outras limitações desse coeficiente estão relacionadas em EMBRECHTS *et al.* (1999).

3.2.2. Medidas de concordância

Dadas as limitações do coeficiente de correlação linear, outras medidas de dependência foram propostas, entre elas as baseadas no conceito de concordância:

Sejam (x_1, y_1) e (x_2, y_2) duas observações de variáveis aleatórias (X, Y) . Então, (x_1, y_1) e (x_2, y_2) são ditas concordantes se $(x_1 - x_2)(y_1 - y_2) > 0$ e discordantes se $(x_1 - x_2)(y_1 - y_2) < 0$. Medidas tais como o tau de Kendall e o rho de Spearman, apresentadas nas seções a seguir, são definidas em função dessas relações de concordância.

3.2.2.1. Tau de Kendall

Esta medida é definida como a probabilidade de concordância menos a probabilidade de discordância de n observações de um vetor (X, Y) de variáveis aleatórias contínuas:

$$\tau = \tau_{XY} = P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0] \quad (3-12)$$

Teorema 3.5 (Cópula baseada no tau de Kendall)

O teorema 5.1.3 de NELSEN (2006) estabelece: sejam X e Y variáveis aleatórias com cópula C . Então, a versão de população do tau de Kendall para X e Y é dada por:

$$\tau_{XY} = \tau_C = -1 + 4 \iint_{[0,1]^2} C(u, v) dC(u, v) \quad (3-13)$$

NELSEN (2006) ainda observa que a integral dupla de (3-13) é o valor esperado da função $C(U, V)$, onde U e V são variáveis aleatórias da distribuição uniforme padrão e cuja distribuição conjunta é C , ou seja:

$$\tau_C = -1 + 4E(C(U, V)) \quad (3-14)$$

3.2.2.2. Rho de Spearman

Sejam (X_1, Y_1) , (X_2, Y_2) e (X_3, Y_3) três amostras independentes de variáveis aleatórias com distribuição conjunta H (cujas marginas são F e G) e cópula C . A versão de população da medida Rho de Spearman é definida como proporcional à probabilidade de concordância menos a probabilidade de discordância para os vetores (X_1, Y_1) e (X_2, Y_3) através da equação:

$$\rho_{S_{XY}} = 3(P[(X_1 - X_2)(Y_1 - Y_3) > 0] - P[(X_1 - X_2)(Y_1 - Y_3) < 0]) \quad (3-15)$$

Segundo a definição, os dois vetores aleatórios possuem as mesmas distribuições marginais, mas enquanto o primeiro possui distribuição conjunta H , os componentes do segundo são independentes.

Teorema 3.6 (Cópula baseada no rho de Spearman)

O teorema 5.1.6 de NELSEN (2006) estabelece: sejam X e Y variáveis aleatórias com cópula C . Então, a versão de população do Rho de Spearman para X e Y é dada por:

$$\rho_{S_{XY}} = \rho_{S_C} = 12 \iint_{[0,1]^2} C(u, v) dudv - 3 \quad (3-16)$$

3.2.3. Dependência nas caudas

Essa medida é muito útil na modelagem de eventos com dependência nos valores extremos das variáveis aleatórias. Geometricamente, concentra-se nas caudas dos quadrantes direito superior e esquerdo inferior de uma distribuição conjunta bivariada. Entretanto, para o caso de dimensões maiores, a avaliação da dependência nas caudas não é tão simples.

Conforme JOE (1997), o coeficiente de dependência na cauda superior entre as variáveis aleatórias X e Y , com distribuições marginais F e G , respectivamente, é dado por:

$$\lambda_U = \lim_{\alpha \rightarrow 1^-} P[Y > G^{-1}(\alpha) | X > F^{-1}(\alpha)], \quad (3-17)$$

dado que o limite $\lambda_U \in [0, 1]$ existe. Se $\lambda_U \in (0, 1]$, então X e Y são assintoticamente dependentes na cauda superior. Se $\lambda_U = 0$, as mesmas são assintoticamente independentes na cauda superior. De modo semelhante, a dependência na cauda inferior é dada por:

$$\lambda_L = \lim_{\alpha \rightarrow 0^+} P[Y \leq G^{-1}(\alpha) | X \leq F^{-1}(\alpha)], \quad (3-18)$$

dado que o limite $\lambda_L \in [0, 1]$ existe. Se $\lambda_L \in (0, 1]$, então X e Y são assintoticamente dependentes na cauda inferior. Se $\lambda_L = 0$, as mesmas são assintoticamente independentes na cauda inferior.

Teorema 3.7 (Dependência nas caudas em função da cópula)

NELSEN (2006), no teorema 5.4.2, demonstra que os coeficientes de dependência caudal λ_U e λ_L são não paramétricos e dependem somente da cópula C das variáveis aleatórias X e Y . Se os limites de (3-17) e (3-18) existem, então:

$$\lambda_U = 2 - \lim_{t \rightarrow 1^-} \frac{1 - C(t, t)}{1 - t} \quad (3-19)$$

$$\lambda_L = \lim_{\alpha \rightarrow 0^+} \frac{C(t,t)}{t} \quad (3-20)$$

Uma vez que a função caudal depende exclusivamente da cópula de X e Y , a mesma é, portanto, invariante a transformações estritamente crescentes em X e Y .

3.3. Cópulas elípticas

Embora exista uma grande variedade de cópulas (JOE, 1997; NELSEN, 2006), poucas são tratáveis. Destarte, na modelagem de dependências, a escolha frequentemente recai em cópulas elípticas ou em arquimedianas.

Uma cópula elíptica é obtida do teorema de Sklar, especificamente quando a função de distribuição F , na relação (3-2), é elíptica. Desse modo, cópulas elípticas também são simétricas nas caudas.

Distribuições elípticas consistem em generalizações de distribuições normais multivariadas. Consequentemente, compartilham de muitas propriedades importantes das distribuições normais, o que permite, por exemplo, a simulação de distribuições elípticas a partir de transformações lineares sobre a distribuição elíptica padrão.

Definição 3.4 (Distribuição elíptica)

Seja $\mathbf{X} = (X_1, X_2, \dots, X_n)$ um vetor de variáveis aleatórias. \mathbf{X} possui distribuição elíptica multivariada com parâmetros μ , Σ , ψ , denotada por $\mathbf{X} \sim E_n(\mu, \Sigma, \psi)$, se a função característica correspondente puder ser expressa por:

$$\varphi_{\mathbf{X}}(t) = \exp(it^T \mu) \psi \left(\frac{1}{2} t^T \Sigma t \right), \quad (3-21)$$

onde, $\mu \in \mathbb{R}^n$: um vetor coluna;

Σ : uma matriz definida positiva $n \times n$;

$\psi(t) \in \Psi_n$: alguma função chamada de geradora característica.

Se $X \sim E_n(\mu, \Sigma, \psi)$ tiver uma densidade, a mesma terá a seguinte forma:

$$f_{\mathbf{X}}(x) = \frac{c_n}{\sqrt{|\Sigma|}} g_n \left[\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right], \quad (3-22)$$

onde $g_n(\cdot)$ é uma função geradora de densidade e c_n é uma constante de normalização expressa por:

$$c_n = \frac{\Gamma(n/2)}{(2\pi)^{n/2}} \left(\int_0^\infty x^{\frac{n}{2}-1} g_n(x) dx \right)^{-1}, \text{ com } \int_0^\infty x^{\frac{n}{2}-1} g_n(x) dx < \infty$$

A partir da definição acima, LANDSMAN & VALDEZ (2003) mostraram que distribuições elípticas e suas marginais são invariantes a transformações lineares, isto é, qualquer combinação linear de variáveis aleatórias distribuídas elípticamente resulta em outra distribuição elíptica.

LANDSMAN & VALDEZ (2003) também mostraram que quando $g(u) = e^{-u}$, o vetor \mathbf{X} assume uma distribuição normal multivariada com média μ e $\Sigma = \text{Cov}(X)$. Neste caso também, segundo EMBRECHTS *et al.* (2003), $\rho(X_i, X_j) = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$. Por isto, o coeficiente de correlação linear é uma medida natural da dependência entre variáveis aleatórias elípticamente distribuídas.

Uma das limitações das distribuições elípticas está no fato das marginais serem todas do mesmo tipo. Uma possível solução consiste em utilizar uma cópula elíptica para juntar distribuições marginais diferentes. Por exemplo, a figura 3.2 ilustra o gráfico de dispersão de 500 amostras simuladas de uma cópula Gaussiana bivariada com $\rho = -0,1$ e os histogramas das distribuições marginais Gamma(2, 1) e t_5 .

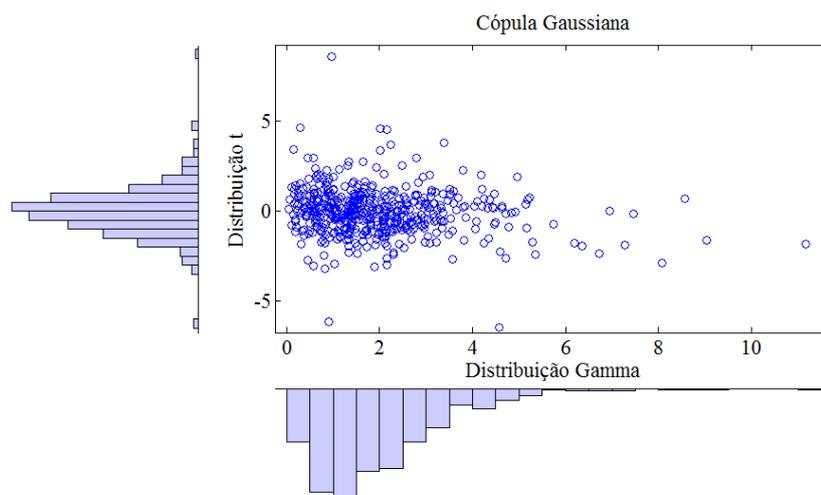


Figura 3.2 - 500 observações de cópula Gaussiana bivariada ($\rho = -0,1$) com distribuições marginais Gamma(2, 1) e t_5 .

Entretanto, nesse tipo de solução, as componentes da matriz Σ não podem ser estimadas através de variâncias e covariâncias. EMBRECHTS *et al.* (2003) propuseram um estimador robusto, função do tau de Kendall, podendo ser utilizado, em todos os casos, sejam as distribuições marginais elípticas ou não:

$$\tau(X_i, X_j) = \frac{2}{\pi} \arcsin(R_{ij}), \quad (3-23)$$

onde $R_{ij} = \rho(X_i, X_j) = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$.

Outra distribuição elíptica importante, a distribuição t-Student, é originada quando o gerador de densidade é especificado em:

- $g_n(x) = \left(1 + \frac{x}{k_p}\right)^{-p}$, onde $p > \frac{n}{2}$ e k_p é uma constante que depende somente de p . A cópula desta distribuição será apresentada brevemente na seção 3.3.2.

3.3.1. Cópula Gaussiana

Utilizando a relação (3-2), obtém-se a expressão geral da cópula Gaussiana:

$$C_R^{Ga}(\mathbf{u}) = \Phi_R^n(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n)), \quad (3-24)$$

onde Φ_R^n é a cópula de $\mathbf{X} \sim \mathcal{N}_d(0, R)$ – distribuição multivariada normal padrão com matriz de correlação linear $R_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$ – e Φ^{-1} , a inversa (função quantil) da distribuição normal univariada. Para o caso bivariado, tem-se:

$$C_R^{Ga}(u_1, u_2) = \int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} \frac{1}{2\pi(1-R_{12}^2)} \exp\left\{-\frac{x^2+y^2-2R_{12}xy}{2(1-R_{12}^2)}\right\} dx dy, \quad (3-25)$$

onde $R_{12} = \rho$ é o coeficiente de correlação linear. Gráficos de dispersão com diferentes parâmetros são visualizados na figura 3.3.

Cópulas Gaussianas, sendo elípticas, possuem simetria radial com coeficientes de caudas superior e inferior iguais. Portanto, não possuem dependência nas caudas e, por isso, não são indicadas para modelar a relação entre variáveis aleatórias de eventos extremos. Esta característica pode ser visualizada nos gráficos de densidade e de curvas de densidade da figura 3.4.

Independente da generalização de (3-25) para o caso n -variado, técnicas de simulação (CHERUBINI *et al.*, 2004), aplicadas no algoritmo proposto nesta tese, permitem gerar facilmente variáveis aleatórias de uma cópula elíptica qualquer, até mesmo para aquelas sem expressão analítica definida.

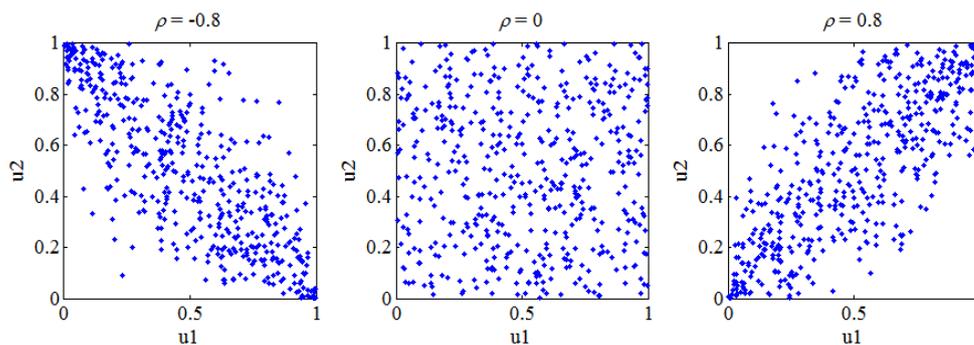


Figura 3.3 - Dispersão de 500 amostras simuladas da cópula Gaussiana bivariada com diferentes parâmetros de dependência: $\rho = -0,8$ (dependência negativa); $\rho = 0$ (independência) e $\rho = 0,8$ (dependência positiva). Os gráficos possuem dupla simetria: (1) em torno da primeira diagonal (devido à permutabilidade)¹ e (2) em torno da segunda diagonal (correspondendo ao fato que cópulas Gaussianas são radialmente simétricas).

Cópula Gaussiana, $\rho = -0.8$

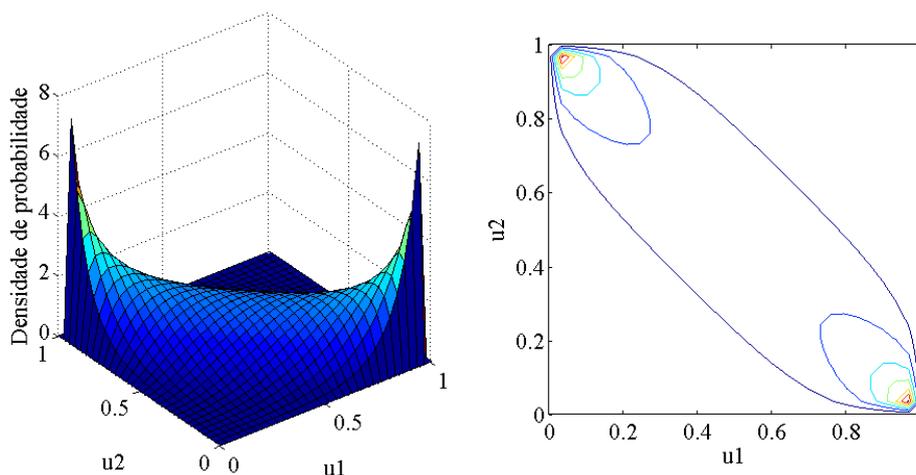


Figura 3.4 - Gráfico de densidade e curvas de densidade da cópula Gaussiana com $\rho = -0,8$, evidenciando a ausência de dependência nas caudas, isto é, nos cantos superior esquerdo e inferior direito do quadrado unitário.

3.3.2. Cópula t-Student

Outra cópula elíptica muito aplicada em finanças é a t-Student. Fazendo uso novamente da relação (3-2), obtém-se a expressão geral da cópula t-Student:

$$C_{v,R}^t(\mathbf{u}) = t_{v,R}^n(t_v^{-1}(u_1), \dots, t_v^{-1}(u_n)), \quad (3-26)$$

onde:

v é o grau de liberdade da distribuição;

$$R_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}} \text{ para } i, j \in \{1, \dots, n\};$$

¹ $C(u_1, u_2) = C(u_2, u_1)$

$t_{v,R}^n$ é a distribuição de $\sqrt{v}\mathbf{Y}/\sqrt{S}$, com $S \sim \chi_v^2$ e $\mathbf{Y} \sim \mathcal{N}_n(0, R)$ independentes;

t_v denota as marginais de $t_{v,R}^n$, isto é, a distribuição de $\sqrt{v}Y_1/\sqrt{S}$.

No caso bivariado, a expressão da cópula t-Student é dada por:

$$C_{v,R}^t(u_1, u_2) = \int_{-\infty}^{t_v^{-1}(u_1)} \int_{-\infty}^{t_v^{-1}(u_2)} \frac{1}{2\pi(1-R_{12}^2)^{1/2}} \exp\left\{1 + \frac{x^2 - 2R_{12}xy + y^2}{v(1-R_{12}^2)}\right\}^{-\frac{(v+2)}{2}} dx dy, \quad (3-27)$$

onde $R_{12} = \rho$ é o coeficiente de correlação linear.

Para esta cópula simétrica, os coeficientes de dependência são iguais a:

$$\lambda_U = \lambda_L = 2\bar{F}_{v+1}\left(\sqrt{v+1} \sqrt{\frac{1-R_{12}}{1+R_{12}}}\right), \quad (3-28)$$

A relação (3-28) mostra que o coeficiente de dependência na cauda é crescente em R_{12} e decrescente em v , convergindo para zero quando v tende para infinito. Isto significa que quando v se aproxima de infinito a cópula-t se aproxima da cópula Gaussiana, que não possui dependência nas caudas. A figura 3.5 mostra a dispersão dos dados simulados de uma cópula t-Student com 1 grau de liberdade. Comparativamente aos gráficos da figura 3.3, ainda que os dados tenham sido simulados com a mesma correlação que os da cópula Gaussiana, observa-se que eventos extremos são muito mais prováveis em cópulas t. Adicionalmente, as relações de simetria dessa cópula podem ser visualizadas nos gráficos de densidade e de curvas de densidade da figura 3.6.

De forma semelhante às cópulas Gaussianas, amostras da cópula t-Student para o caso n -variado podem ser obtidas fazendo uso de técnicas de simulação.

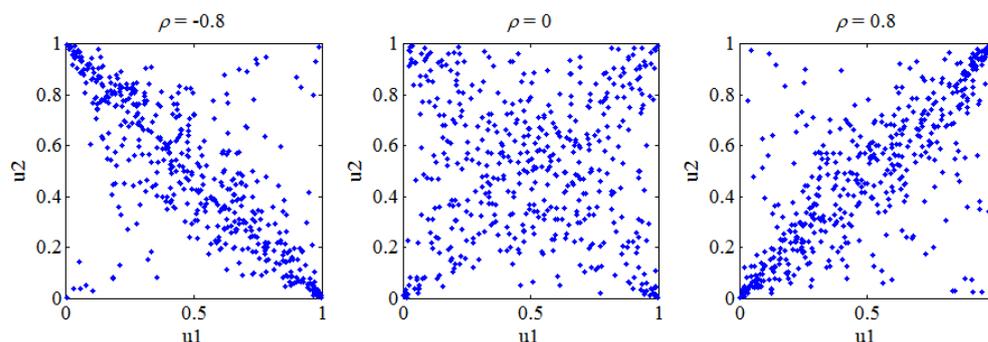


Figura 3.5 - Dispersão de 500 amostras simuladas da cópula t ($v = 1$) com diferentes parâmetros de dependência: $\rho = -0,8$ (dependência negativa); $\rho = 0$ (independência) e $\rho = 0,8$ (dependência positiva). Os gráficos possuem dupla simetria: (1) em torno da primeira diagonal (devido à permutabilidade) e (2) em torno da segunda diagonal (correspondendo ao fato que cópulas t são radialmente simétricas).

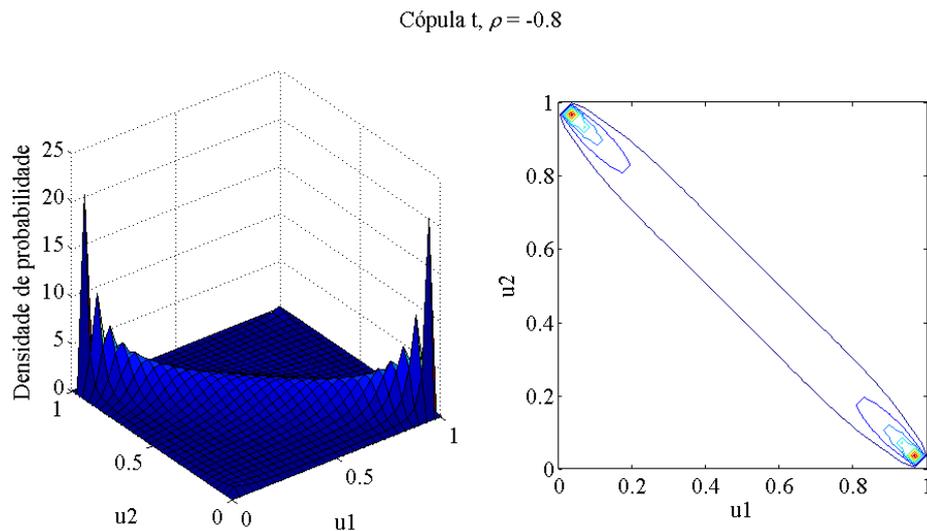


Figura 3.6 - Gráfico de densidade e curvas de densidade da cópula t_1 com $\rho = -0,8$.

3.4. Cópulas arquimedianas

Amplamente utilizadas em análise de riscos financeiros e atuariais (EMBRECHTS *et al.*, 2003), cópulas arquimedianas² possuem propriedades que tornam a sua aplicabilidade conveniente. Além de muitas famílias de cópulas paramétricas serem arquimedianas, todas as subclasses mais comuns destas podem: i) ser expressas de forma explícita e fechada; e ii) representar uma grande variedade de estruturas de dependência, incluindo assimetria e dependência de valores extremos; contornando as principais limitações das cópulas elípticas. No entanto, estes tipos de cópulas não derivam diretamente do teorema de Sklar e, desse modo, é necessário considerar algumas premissas para que essas funções sejam cópulas, de fato.

Nas subseções seguintes, alguns aspectos relevantes sobre cópulas bivariadas e multivariadas serão apresentados. NELSEN (2006) e CHERUBINI *et al.* (2004) constituem boas referências para aprofundar os conceitos introdutórios destas cópulas.

² De acordo com MAI & SCHERER (2012), o termo cópula arquimediana deriva de uma propriedade puramente analítica, relacionada à propriedade arquimediana das normas triangulares (t-norma), sendo utilizado pela primeira vez em LING (1965). Uma t-norma é um operador $T: [0,1] \times [0,1] \rightarrow [0,1]$ crescente, comutativo e associativo que satisfaz $T(x, 1) = x$, $\forall x \in [0,1]$. Em particular, uma t-norma T_f é arquimediana se, e somente se, T_f é contínua em ambos os argumentos e $T_f(x, x) < x$, $\forall x \in [0,1]$.

3.4.1. Cópulas bivariadas

Cópulas arquimedianas são construídas a partir de uma função $\varphi: [0,1] \rightarrow [0, \infty]$, contínua, estritamente decrescente e convexa e tal que $\varphi(1) = 0$. Chamada de função geradora da cópula³, φ é do tipo estrita sempre que $\varphi(0) = +\infty$. Define-se também a pseudoinversa de φ como:

$$\varphi^{[-1]}(t) = \begin{cases} \varphi^{-1}(t), & 0 \leq t \leq \varphi(0), \\ 0, & \varphi(0) \leq t \leq +\infty. \end{cases} \quad (3-29)$$

onde $\varphi^{[-1]}(\varphi(t)) = t, \forall t \in [0, 1]$. Se φ é do tipo estrita, então $\varphi^{[-1]}(u) = \varphi^{-1}(u)$. Assim, dada uma função geradora e sua pseudoinversa, uma cópula arquimediana bivariada estrita é uma função do tipo:

$$C(u_1, u_2) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2)), \quad (3-30)$$

Uma cópula arquimediana C apresenta as seguintes propriedades, muito úteis na extensão multivariada de cópulas bivariadas:

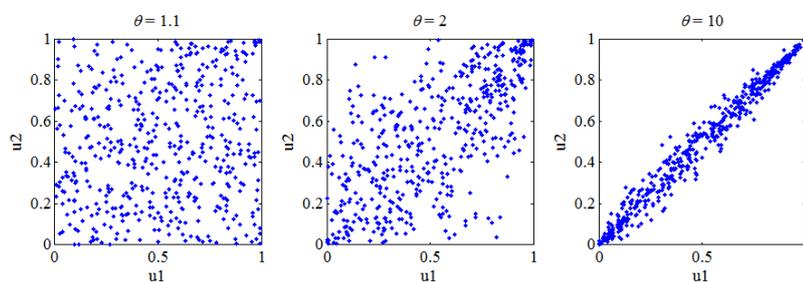
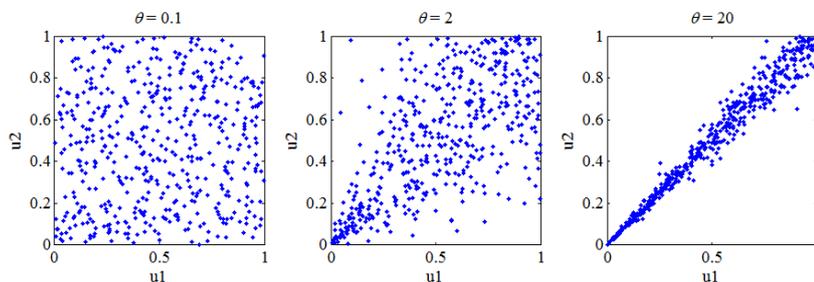
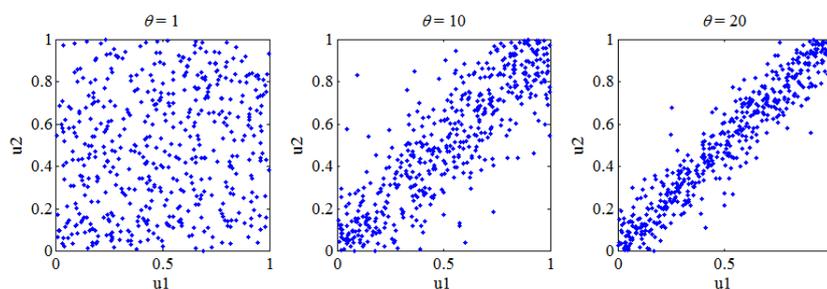
- C é simétrica, isto é, $C(u_1, u_2) = C(u_2, u_1)$
- C é associativa, isto é, $C(C(u_1, u_2), u_3) = C(u_1, C(u_2, u_3))$

A tabela seguinte relaciona algumas das famílias de cópulas arquimedianas bivariadas uniparamétricas mais utilizadas, $C_\theta(u, v)$, com suas respectivas funções geradoras, $\varphi_\theta(t)$. Nessas, a estrutura de dependência é controlada por apenas um parâmetro, θ : na cópula de Gumbel (figura 3.7), a dependência perfeita ocorre quando $\theta \rightarrow \infty$, enquanto que $\theta \rightarrow 0$ implica em independência; nas cópulas de Clayton (figura 3.8) e de Frank (figura 3.9), a dependência perfeita ocorre quando $\theta \rightarrow \infty$, enquanto que $\theta \rightarrow 0$ implica em independência. Em particular, a cópula de Frank bivariada é a única da família arquimediana que possui simetria radial, isto é, a cópula e sua cópula de sobrevivência são as mesmas.

³ A função geradora de uma cópula arquimediana é a função geradora de momentos, utilizada para calcular a média e a variância, além de outros momentos de uma variável aleatória. Sendo a função geradora de momentos um caso particular da transformada de Laplace, cópulas arquimedianas são obtidas a partir de tais transformadas (CHERUBINI *et al.*, 2004). Vide, adiante, seção 4.2.4.3.

| Cópula | $\varphi_{\theta}(t)$ | $C_{\theta}(u, v)$ | $\theta \in$ |
|-----------------|---|---|-------------------------------------|
| Clayton | $\theta^{-1}(t^{-\theta} - 1)$ | $[\max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-1/\theta}$ | $[-1, \infty) \setminus \{0\}$ |
| Gumbel | $(-\ln t)^{\theta}$ | $\exp(-[(-\ln u)^{\theta} + (-\ln v)^{\theta}]^{1/\theta})$ | $[1, \infty)$ |
| Frank | $-\ln \left[\frac{e^{-\theta t} - 1}{e^{-\theta} - 1} \right]$ | $-\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$ | $(-\infty, \infty) \setminus \{0\}$ |
| Joe | $-\ln[1 - (1-t)^{\theta}]$ | $1 - [(1-u)^{\theta} + (1-v)^{\theta} - (1-u)^{\theta}(1-v)^{\theta}]^{1/\theta}$ | $[1, \infty)$ |
| Ali-Mikhail-Haq | $\ln \frac{1 - \theta(1-t)}{t}$ | $\frac{uv}{1 - \theta(1-u)(1-v)}$ | $[-1, 1)$ |

Tabela 3.1 - Funções geradoras e parâmetros de cópulas arquimedianas bivariadas.

Figura 3.7 - Dispersão de 500 amostras simuladas da cópula de Gumbel com diferentes parâmetros de dependência: $\theta \rightarrow 1$ corresponde à independência e $\theta \rightarrow \infty$, dependência perfeita.Figura 3.8 - Dispersão de 500 amostras simuladas da cópula de Clayton com diferentes parâmetros de dependência: $\theta \rightarrow 0$ corresponde à independência e $\theta \rightarrow \infty$, dependência perfeita.Figura 3.9 - Dispersão de 500 amostras simuladas da cópula de Frank com diferentes parâmetros de dependência: $\theta \rightarrow 0$ corresponde à independência e $\theta \rightarrow \infty$, dependência perfeita. O gráfico apresenta também dois eixos de simetria: (1) simetria relativa à primeira diagonal (devido à permutabilidade) e (2) simetria relativa à segunda diagonal, correspondendo ao fato da cópula bivariada de Frank ser radialmente simétrica.

O tau de Kendall pode ser definido em termos da função geradora de uma cópula arquimediana C através da seguinte relação:

$$\tau_C = 1 + 4 \int_0^1 \frac{\varphi(t)}{\varphi'(t)} dt \quad (3-31)$$

A relação (3-29) permite estimar o parâmetro θ da cópula arquimediana. Expressões de θ em função de τ estão tabeladas para mais de vinte cópulas arquimedianas uniparamétricas em NELSEN (2006).

Os coeficientes de dependência nas caudas também podem ser escritos em termos da função geradora da cópula arquimediana:

$$\lambda_U = 2 - 2 \lim_{s \rightarrow 0} \left(\frac{\partial \varphi^{-1}(2s)}{\partial s} / \frac{\partial \varphi^{-1}(s)}{\partial s} \right) \quad (3-32)$$

$$\lambda_L = 2 - 2 \lim_{s \rightarrow +\infty} \left(\frac{\partial \varphi^{-1}(2s)}{\partial s} / \frac{\partial \varphi^{-1}(s)}{\partial s} \right) \quad (3-33)$$

A realização dos cálculos das relações anteriores resulta em expressões fechadas dos coeficientes λ_U e λ_L em função de θ . Na tabela seguinte, estas medidas de dependência são apresentadas juntamente com o tau de Kendall para as famílias da tabela 3.1. É fácil observar, nos gráficos de dispersão, que a dependência da distribuição baseada na cópula de Gumbel (figura 3.10) é mais forte na cauda superior do que na inferior. De modo oposto, a dependência da distribuição baseada na cópula de Clayton (figura 3.11) é mais forte na cauda inferior do que na superior. Por outro lado, a cópula de Frank (figura 3.12) não possui dependência nas caudas.

| Cópula | τ_C | (λ_L, λ_U) |
|-----------------|---|---------------------------------|
| Clayton | $\frac{\theta}{\theta + 2}$ | $(2^{-\frac{1}{\theta}}, 0)$ |
| Gumbel | $1 - \frac{1}{\theta}$ | $(0, 2 - 2^{\frac{1}{\theta}})$ |
| Frank | $1 - \frac{4}{\theta} + 4 \frac{D_1(\theta)^*}{\theta}$ | $(0, 0)$ |
| Joe | $1 + \frac{4}{\theta^2} \int_0^1 x(\ln x)(1-x)^{2(1-\theta)/\theta} dx$ | $(0, 2 - 2^{\frac{1}{\theta}})$ |
| Ali-Mikhail-Haq | $1 - \frac{2(\theta + (1-\theta)^2 \ln(1-\theta))}{3\theta^2}$ | $(0, 0)$ |

* $D_1(\theta) = \int_0^\theta \frac{c/\theta}{\exp(x)-1} dx$ é a função de Debye.

Tabela 3.2 - Parâmetros de dependência de cópulas arquimedianas bivariadas.

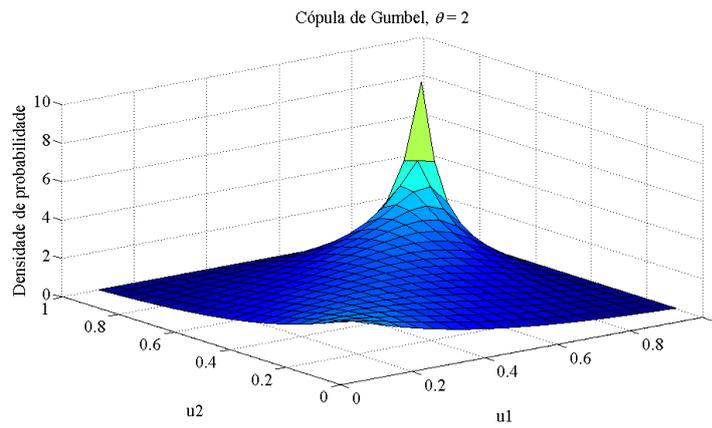


Figura 3.10 - Gráfico de densidade da cópula de Gumbel com $\theta = 2$

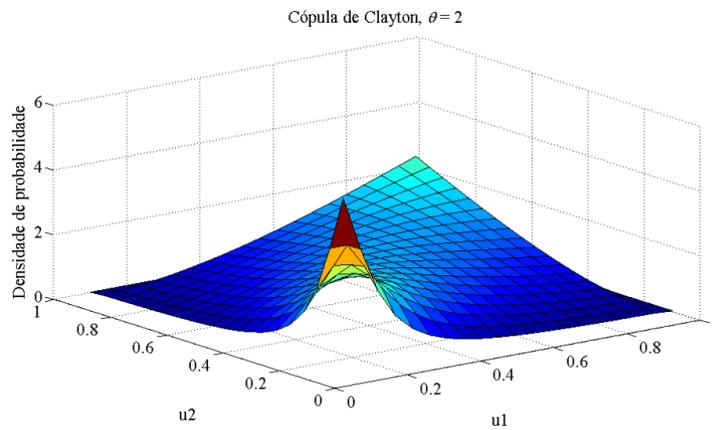


Figura 3.11 - Gráfico de densidade da cópula de Clayton com $\theta = 2$

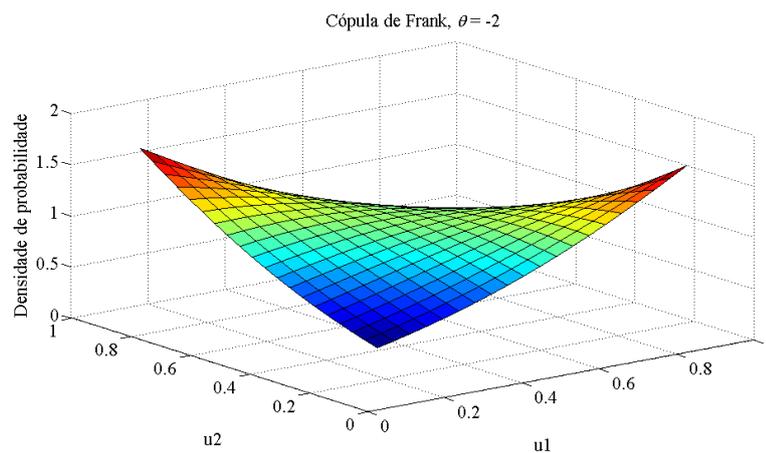


Figura 3.12 - Gráfico de densidade da cópula de Frank com $\theta = -2$

A família de cópulas arquimedianas biparamétricas incluem BB1, BB2, BB3, BB6 e BB7 (notação de JOE, 1997). A família BB7, também conhecida como Joe-Clayton, tem a estrutura de dependência controlada pelos coeficientes λ_U e λ_L :

$$C_{JC}(u_1, u_2) = 1 - (1 - \{[1 - (1 - u_1)^\kappa]^{-\gamma} + [1 - (1 - u_2)^\kappa]^{-\gamma} - 1\}^{1/\gamma})^{1/\kappa}, \quad (3-34)$$

onde $\kappa = 1/\log_2(2 - \lambda_U)$ e $\gamma = 1/\log_2(\lambda_L)$.

3.4.2. Cópulas multivariadas

Apesar da utilidade das cópulas ter sido reconhecida no contexto da modelagem de dados bivariados e, por conseguinte, com a estimação de um único parâmetro de dependência, novas aplicações têm motivado o não trivial desenvolvimento de cópulas multivariadas arquimedianas. Basicamente, a literatura apresenta dois tipos principais de modelos para construção destas: os arquimedianos hierárquicos clássicos ou aninhados e os baseados em *pair-copulas*. Os primeiros realizam expansões multivariadas de cópulas arquimedianas bivariadas; enquanto os últimos empregam blocos construtivos simples, as *pair-copulas*, obtidos a partir da decomposição hierárquica de uma cópula multivariada.

Os modelos arquimedianos hierárquicos clássicos sucederam a chamada construção permutável (JOE, 1997; EMBRECHTS *et al.*, 2003) que expandia cópulas bivariadas para dimensões maiores através de uma generalização da relação (3-30), sendo expressa por:

$$C^n(\mathbf{u}) = \varphi^{-1}(\varphi(u_1) + \dots + \varphi(u_n)) \quad (3-35)$$

onde $\mathbf{u} = (u_1, \dots, u_n)$ é um vetor n -dimensional com $u_k \in [0, 1]$ para $k = 1, \dots, n$; φ e φ^{-1} são conforme definidas em (3-29). No caso tridimensional, fazendo uso das propriedades associativa e de simetria das cópulas,

$$\begin{aligned} C^3(\mathbf{u}) &= \varphi^{-1}\left(\varphi \circ \varphi^{-1}\left(\varphi \circ \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) + \varphi(u_3)\right) + \varphi(u_3)\right) \\ &= C(C(u_1, u_2), u_3) \end{aligned} \quad (3-36)$$

Ocorre que esta construção é extremamente restritiva porque, além da premissa de monotonicidade de φ^{-1} em \mathbb{R}_+ (de outro modo a cópula não é arquimediana) (CHERUBINI *et al.*, 2004), requer também dependência simétrica entre todos os pares de \mathbf{u} , isto é, entre (u_1, u_2) , (u_2, u_3) , (u_1, u_3) , devido haver um único parâmetro de dependência. Esta restrição se torna mais onerosa à medida que

crece o número de distribuições marginais. Com isso, esta construção tem a grande desvantagem de não permitir modelar separadamente a dependência entre os pares.

As limitações da construção permutável foram, em parte, resolvidas por JOE (1997) com o modelo arquimediano completamente aninhado (*fully nested archimedean*), que continuaria a realizar expansões multivariadas de cópulas bivariadas, mas, dessa vez, permitiria graus diferentes de dependência em marginais bivariadas distintas, pois existe um gerador característico para cada par de variáveis aleatórias. Um exemplo tridimensional dessa construção pode ser representado analiticamente por:

$$\begin{aligned} C(u_1, u_2, u_3) &= C_{21}(u_3, C_{11}(u_1, u_2)) = \\ &= \varphi_{21}^{-1} \left(\varphi_{21}(u_3) + \varphi_{21} \circ \varphi_{11}^{-1}(\varphi_{11}(u_1) + \varphi_{11}(u_2)) \right), \end{aligned} \quad (3-37)$$

ou, através do diagrama da figura 3.13. Esta representação ilustra a construção em níveis/hierarquia ou por agregação de blocos/ninhos de geradores, que justifica a nomenclatura adotada para esta técnica de construção de cópulas multivariadas. Na mesma, a cada passo, ou nível hierárquico, é adicionada uma dimensão (variável). Os nós u_1 e u_2 são acoplados através da cópula C_{11} , constituindo o nível 1; os nós u_3 e $C_{11}(u_1, u_2)$ são acoplados através da cópula C_{21} , no nível 2. Assim, a cópula para o caso de 3 dimensões requer 2 cópulas bivariadas, C_{11} e C_{21} , e geradoras respectivas, φ_{11} e φ_{21} .

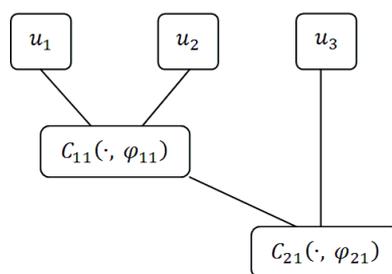


Figura 3.13 - Cópula arquimediana tridimensional aninhada.

Adicionalmente, o modelo completamente aninhado ainda permite que as cópulas sejam arquimedianas de famílias diferentes.

Para o caso n -dimensional, a expressão correspondente é dada por:

$$C(u_1, \dots, u_n) = \varphi_{n-1,1}^{-1} \left\{ \varphi_{n-1,1}(u_n) + \varphi_{n-1,1} \circ \varphi_{n-2,1}^{-1} \left\{ \varphi_{n-2,1}(u_{n-1}) + \varphi_{n-2,1} \circ \dots \circ \varphi_{11}^{-1} (\varphi_{11}(u_1) + \varphi_{11}(u_2)) \right\} \right\}, \quad (3-38)$$

Construções alternativas, também propostas por JOE (1997), foram posteriormente discutidas. Entre elas, a parcialmente aninhada (WHELAN, 2004; MCNEIL, 2008) e a hierarquicamente aninhada (WHELAN, 2004; SAVU & TREDE, 2006) que diferem da completamente aninhada pelo modo que agrupam as variáveis no primeiro nível da hierarquia. Por outro lado, todos os modelos hierárquicos precisam atender a algumas premissas para garantir que a cópula do último nível seja uma função de distribuição: i) o número de cópulas deve decrescer a cada nível, de modo que o último nível tenha uma única cópula; ii) todas as funções geradoras e as inversas correspondentes precisam ser completamente monótonas; iii) os parâmetros de dependência devem decrescer a cada nível.

Propostos no mesmo período dos modelos arquimedianos aninhados (JOE, 1997) e, estendidos em AAS *et al.* (2007), os modelos *pair-copulas* também são hierárquicos e mais flexíveis na construção de cópulas multivariadas, pois não existem as restrições de parâmetros citadas no parágrafo anterior e, além disso, a construção admite quaisquer classes de cópulas, não somente as arquimedianas. Desse modo, todos os tipos e forças de dependência podem ser obtidos com modelos *pair-copulas*.

Modelos *pair-copulas* são baseados na decomposição hierárquica de uma cópula n -variada em uma cascata de cópulas bivariadas potencialmente diferentes. Para entender esse mecanismo, sejam F uma distribuição conjunta com densidade h e F_1, \dots, F_n suas distribuições marginais estritamente contínuas com densidades f_i , conforme MENDES (2009). Qualquer função de densidade multivariada pode ser decomposta da seguinte forma:

$$h(x_1, \dots, x_n) = f(x_n) \cdot f(x_{n-1}|x_n) \cdot f(x_{n-2}|x_{n-1}, x_n) \cdots f(x_1|x_2, \dots, x_n) \quad (3-39)$$

As densidades condicionais de (3-35) podem ser escritas como funções de densidades de cópulas correspondentes. Assim, para todo j ,

$$f(x|v_1, v_2, \dots, v_n) = c_{xv_j|v_{-j}} \left(F(x|v_{-j}), F(v_j|v_{-j}) \right) \cdot f(x|v_{-j}) \quad (3-40)$$

onde v_{-j} denota o vetor n -dimensional \mathbf{v} sem a componente j . $c_{xv_j|v_{-j}}(\cdot, \cdot)$ é uma densidade marginal da cópula bivariada ou, em outras palavras, é uma *pair-copula*.

Expressando todas as densidades condicionais de (3-39) através de (3-40), verifica-se que $f(x_1, \dots, x_n)$ é fatorada, exclusivamente, em termos de distribuições marginais e cópulas bivariadas. Portanto, a densidade multivariada pode ser expressa como um produto de *pair-copulas*, atuando em várias distribuições de probabilidade distintas. Ademais, dada a natureza iterativa dessa construção, em uma fatoração específica, muitas reparametrizações diferentes são feitas, garantindo a flexibilidade desse método de construção de cópulas multivariadas.

Uma vez que a construção *pair-copula* envolve distribuições marginais condicionais, $F(x|\mathbf{v})$, JOE (1997) mostrou que as mesmas consistem na derivada parcial com respeito ao segundo argumento da cópula condicional, isto é:

$$F(x|\mathbf{v}) = \frac{\partial C_{x,v_j|\mathbf{v}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))}{\partial F(v_j|\mathbf{v}_{-j})} \quad (3-41)$$

Para um caso especial em que v é univariado e x e y são variáveis do tipo uniforme padrão, (3-41) é simplificada a:

$$F(x|v) = \frac{\partial C_{xv}(x,v,\Theta)}{\partial v} \quad (3-42)$$

onde Θ é o conjunto de parâmetros da cópula.

À medida que o número de variáveis aumenta, cresce o número de cópulas bivariadas. Por exemplo, para $n = 5$, há 240 diferentes decomposições possíveis. Em função disto, BEDFORT & COOKE (2001) e KUROWICKA & COOKE (2001) desenvolveram modelos gráficos, denominados de *vines* regulares, para obter, de modo sistemático, as possíveis decomposições *pair-copulas*. Casos especiais desses modelos gráficos incluem o C-vine (hierárquico canônico) e o D-vine. Cada um deles fornece um modo de arranjar as *pair-copulas*, em uma estrutura de árvore (hierárquica), facilitando a análise das múltiplas dependências e, portanto, a modelagem multidimensional. Para uma descrição detalhada de *vines* regulares, consultar AAS *et al.* (2007).

3.5. Métodos de estimação de parâmetros

Uma etapa essencial na modelagem por cópulas é a determinação dos parâmetros de dependência e das distribuições marginais. Muito embora o mais comumente utilizado seja o estimador de máxima verossimilhança (JOE, 1997), a literatura descreve outros métodos ainda não totalmente explorados. Esta seção apresentará brevemente as principais técnicas de estimação.

3.5.1. Estimação por máxima verossimilhança

Também conhecido como método de um passo ou de estimação por máxima verossimilhança (MLE – *Maximum Likelihood Estimation*), é utilizado em modelos completamente paramétricos, nos quais os parâmetros das distribuições marginais e da cópula são estimados simultaneamente através da maximização da função log-verossimilhança.

Seja $\theta \in \Theta$ um vetor de parâmetros a estimar. Este vetor pode ser dividido entre os parâmetros das marginais e da cópula: $\theta = [\varphi', \gamma', \delta']$, onde $\varphi' \in \phi$ denota o(s) parâmetro(s) de $f(x)$; $\gamma' \in \Gamma$ representa o(s) parâmetro(s) de $g(y)$ e $\delta' \in \Delta$ denota o(s) parâmetro(s) de $C(F(x), G(x))$. Amostrando as variáveis x_t e y_t em $t = 0, \dots, T$, representa-se a densidade conjunta (utilizando a relação (3-7)) e a função de log-verossimilhança L_{XY} resultante por:

$$h(x_t, y_t, \theta) = f(x_t; \varphi) \cdot g(y_t; \gamma) \cdot c(F(x_t; \varphi); G(y_t; \gamma); \delta) \quad (3-43)$$

$$\begin{aligned} L_{XY} &= \sum_{t=1}^T \log(f(x_t; \varphi)) \cdot \sum_{t=1}^T \log(g(y_t; \gamma)) \times \sum_{t=1}^T \log(c(F(x_t; \varphi); G(y_t; \gamma); \delta)) \\ &= L_X(\varphi) + L_Y(\gamma) + L_C(\varphi, \gamma, \delta) \end{aligned} \quad (3-44)$$

Então, o estimador de máxima verossimilhança (ML) é dado por:

$$\hat{\theta} = \arg \max L_{XY} \quad (3-45)$$

MLE é um método consistente, assintoticamente normal e ótimo, pois, entre todos os estimadores bem comportados, o MLE possui a menor variância para grandes amostras. Entretanto, no caso de cópulas arquimedianas hierárquicas, devido à complexidade de suas densidades, uma expressão fechada para o MLE pode não ser obtida. Neste caso, é necessário aplicar algoritmos de otimização para

encontrar a densidade n -dimensional (SAVU & TREDE, 2006), tornando a estimação muito onerosa em termos de custo computacional.

3.5.2. Método de inferência para as marginais

Introduzido por JOE & SU (1996), o método de inferência para as marginais (IFM – *Inference Functions for Margins*) estima os parâmetros em duas etapas. Na primeira, os parâmetros φ e γ das densidades marginais são estimados por MLE:

$$\hat{\varphi} = \arg \max L_X$$

$$\hat{\gamma} = \arg \max L_Y$$

Na segunda etapa, fazem uso desses valores para transformar as marginais em variáveis do tipo uniforme padrão, estimando o(s) parâmetro(s) δ através da maximização da densidade da cópula, isto é:

$$\hat{\delta} = \arg \max L_C(\hat{\varphi}, \hat{\gamma}, \delta)$$

JOE (1997) argumenta que o método IFM é bastante eficiente, uma vez que é totalmente baseado na estimação por máxima verossimilhança, demandando, porém, menor esforço computacional que o MLE. A eficiência relativa entre as estimativas pode ser verificada tanto pela comparação das matrizes de covariância assintóticas, quanto pela comparação de seus erros quadráticos médios das simulações.

3.5.3. Método semiparamétrico

Originalmente sugerido por OAKES (1994) e formalizado por GENEST *et al.* (1995), a estimação por máxima verossimilhança canônica (CML – *Canonical Maximum Likelihood*) é considerada um padrão para estimação de parâmetros de cópulas baseadas em *rank*. Consiste em uma versão modificada da estimação por máxima verossimilhança (MLE), na qual as distribuições marginais desconhecidas são substituídas por distribuições empíricas equivalentes. Neste caso, os parâmetros das marginais são estimados da distribuição empírica, de modo não paramétrico.

A função de distribuição empírica é definida por:

$$\hat{F}(\cdot) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\{X_t \leq \cdot\}}$$

onde $\{X_t \leq \cdot\}$ é a função indicadora. O parâmetro δ da cópula pode ser estimado, então, através da maximização de função de log-verossimilhança da densidade da cópula, utilizando as variáveis transformadas:

$$L_C(\delta) = \sum_{t=1}^T \log \left(c \left(\hat{F}(x_t); \hat{G}(y_t) \right) \right) = \sum_{t=1}^T \log(c(\hat{u}_t; \hat{v}_t))$$

Dessa forma, o estimador semiparamétrico é expresso por:

$$\hat{\delta} = \arg \max L_C(\delta)$$

Sob condições fracas de regularidade, este estimador é consistente e assintoticamente gaussiano (GENEST *et al.*, 1995). Na prática, entretanto, a CML pode requerer computação intensiva. Além disso, sua aplicação limita-se aos casos em que a densidade da cópula segue a medida de Lebesgue (GENEST *et al.*, 2013).

3.5.4. Métodos não paramétricos

Nos casos em que uma cópula não tem uma densidade ou quando a maximização da função de log-verossimilhança requer um grande esforço computacional, os métodos não paramétricos constituem uma boa alternativa. Os métodos mais comuns dessa classe são os que utilizam cópulas empíricas, construídas a partir dos dados da amostra, e os baseados no método dos momentos.

Sejam $\{(x_k, y_k)\}_{k=1}^T$ amostras de tamanho T de uma distribuição contínua bivariada. A cópula empírica (DEHEUVELS, 1979) referente a estas amostras é dada por:

$$\hat{C} \left(\frac{i}{T}, \frac{j}{T} \right) = \frac{\text{número de pares } (x_t, y_t) \text{ na amostra tal que } x_t \leq x_{(i)} \text{ e } y_t \leq y_{(j)}}{T}, \quad (3-46)$$

para $i, j = 1, \dots, T$, onde $x_{(i)}$ e $y_{(j)}$ são as estatísticas de ordem i e j de x e y , respectivamente.

DEHEUVELS (1979) também definiu a frequência ou densidade da cópula empírica. Para o caso bivariado:

$$\hat{c} \left(\frac{i}{T}, \frac{j}{T} \right) = \begin{cases} \frac{1}{T}, & \text{se } (x_{(i)}, y_{(j)}) \text{ é um elemento da amostra} \\ 0, & \text{caso contrário} \end{cases} \quad (3-47)$$

A relação entre a cópula empírica bivariada e sua frequência é dada por:

$$\hat{C}\left(\frac{i}{T}, \frac{j}{T}\right) = \sum_{i_1=1}^i \sum_{j_1=1}^j \hat{c}\left(\frac{i_1}{T}, \frac{j_1}{T}\right), \quad (3-48)$$

O conceito de cópulas empíricas pode ser utilizado para definir a versão amostral de muitas medidas de dependência. Por exemplo, o coeficiente de correlação baseado em postos de Kendall, ρ_K , pode ser estimado empiricamente a partir de:

$$\hat{\rho}_K = \frac{4}{T^2-1} \sum_{i=1}^T \sum_{j=1}^T \hat{C}\left(\frac{i}{T}, \frac{j}{T}\right) \cdot \hat{c}\left(\frac{i}{T}, \frac{j}{T}\right) - \frac{ij}{T^2}, \quad (3-49)$$

No entanto, a grande utilidade da cópula empírica, nesta tese, será a de fornecer as distribuições marginais para outra cópula que efetivamente realizará a modelagem probabilística do EDA.

Outra forma de estimação não paramétrica é a do método dos momentos. Entre estes, o da inversão do tau de Kendall (GENEST & RIVEST, 1993) é o mais comum. Em analogia ao método dos momentos na estatística clássica, estimar θ pela inversão do tau de Kendall consiste em resolver a equação da relação (3-14) em θ . Esta estimação está prontamente disponível para as cópulas arquimedianas bivariadas, através de expressões analíticas fechadas (NELSEN, 2006). BORKOWF (2002) mostra que este estimador é consistente e assintoticamente gaussiano sob condições fracas de regularidade.

3.6.

Algoritmos de estimação baseados em cópulas (CEDAs)

O capítulo anterior apresentou algoritmos de estimação de distribuição (EDAs) como métodos estocásticos de otimização, da família dos algoritmos evolutivos, caracterizados pelo uso de modelos probabilísticos na exploração do espaço de busca. Foi visto também que a estimação destes modelos constitui etapa determinante para o desempenho de EDAs.

Uma vez que funções cópulas, de acordo com a teoria exposta neste capítulo, são capazes de capturar toda a estrutura de dependência de um conjunto de variáveis aleatórias multidimensionais – realizando modelagens em problemas importantes de diversas áreas do conhecimento – é bastante factível associá-las a EDAs, através dos algoritmos de estimação baseados em cópulas (CEDAs), com a finalidade de construir distribuições de busca mais apropriadas e de modo mais simplificado que

outros EDAs. Basicamente, cópulas atuam em EDAs na etapa de aprendizagem do algoritmo, isto é, logo após selecionar os melhores indivíduos (vide figura 2.4): inicialmente, na estimação das distribuições marginais e, em seguida, na estimação da estrutura de dependência.

A utilização de cópulas em algoritmos evolutivos é bem recente, sendo apresentado na próxima subseção um levantamento do que já foi produzido. As informações da mesma são baseadas em GONZÁLEZ-FERNÁNDEZ & SOTO (2014) que disponibiliza um pacote em linguagem R dos principais CEDAs publicados.

3.6.1. Revisão dos CEDAs existentes na literatura

O primeiro trabalho que associou cópulas a EDAs foi um relatório técnico (SOTO *et al.*, 2007). Desde então, diversos CEDAs têm sido propostos (WANG *et al.*, 2009a,b; SALINAS-GUTIERREZ *et al.*, 2009; GAO, 2009; SOTO & GONZÁLEZ-FERNÁNDEZ, 2010; SALINAS-GUTIÉRREZ *et al.*, 2010; CUESTA-INFANTE *et al.*, 2010; YE *et al.*, 2010; GONZÁLEZ-FERNÁNDEZ, 2011; SALINAS-GUTIÉRREZ *et al.*, 2011).

Identificada como uma abordagem emergente em problemas de otimização numérica (HAUSCHILD & PELIKAN, 2011), CEDAs assumem usualmente uma distribuição particular para as marginais (normal ou beta, por exemplo) e estimam seus parâmetros através de máxima verossimilhança (SOTO *et al.*, 2007; WANG *et al.*, 2009a,b; SALINAS-GUTIERREZ *et al.*, 2009; SOTO & GONZÁLEZ-FERNÁNDEZ, 2010; SALINAS-GUTIÉRREZ *et al.*, 2010; CUESTA-INFANTE *et al.* 2010; YE *et al.*, 2010; GONZÁLEZ-FERNÁNDEZ, 2011; SALINAS-GUTIÉRREZ *et al.*, 2011). Em outros casos, foram utilizados o método de estimativa de densidade de kernel – KDE (SOTO *et al.*, 2007; GAO, 2009; SALINAS-GUTIÉRREZ *et al.*, 2011) ou distribuições marginais empíricas (CUESTA-INFANTE *et al.*, 2010).

GONZÁLEZ-FERNÁNDEZ & SOTO (2014), na implementação dos algoritmos do pacote em R, organizam CEDAs em dois tipos: baseados em cópulas multivariadas e baseados em fatoração de cópulas. Essa classificação busca separar os algoritmos segundo os métodos de construção de cópulas multivariadas, mas

verifica-se que, no segundo grupo, os autores misturam as arquimedianas aninhadas com as construídas através de *pair-copulas*.

EDAs baseados em cópulas multivariadas

As propostas de EDAs baseados em cópulas multivariadas têm feito uso da cópula normal (SOTO *et al.*, 2007; WANG *et al.*, 2009b) e das arquimedianas (WANG *et al.*, 2009a; GAO, 2009). Estas envolviam expansões multivariadas de cópulas arquimedianas bivariadas através de construções permutáveis, isto é, com uma única função geradora. Por outro lado, a diferença de abordagem entre SOTO *et al.* (2007) e WANG *et al.* (2009b) reside na forma como cada um deles estima as distribuições marginais.

Dois EDAs baseados em cópulas multivariadas foram implementados no pacote em R de GONZÁLEZ-FERNÁNDEZ & SOTO (2014):

- UMDA (*Univariate Marginal Distribution Algorithm*) para variáveis contínuas. Muito embora, não seja originalmente definido em termos de cópulas, este algoritmo é descrito como um CEDA que modela a dependência entre variáveis através do produto de cópulas multivariadas, portanto, supondo independência entre as variáveis;
- GCEDA (*Gaussian Copula Estimation of Distribution Algorithm*). Este algoritmo é descrito como um EDA baseado em cópula normal multivariada, que permite a construção de distribuições com estrutura de dependência normal e marginais não normais.

EDAs baseados em fatoração de cópulas

Alguns EDAs baseados em cópulas construídas através de modelos hierárquicos clássicos têm sido publicados. Estes incluem: SALINAS-GUTIERREZ *et al.* (2009); SOTO & GONZÁLEZ-FERNÁNDEZ (2010); SALINAS-GUTIÉRREZ *et al.* (2010); CUESTA-INFANTE *et al.* (2010); YE *et al.* (2010) e GONZÁLEZ-FERNÁNDEZ (2011). Particularmente, o CEDA de SALINAS-GUTIERREZ *et al.* (2009), implementado no pacote em R, é uma extensão do MIMIC (*Mutual Information Maximization for Input Clustering*) para o domínio contínuo que, novamente, apesar de não ser originalmente definido para cópulas, é descrito como um EDA que modela a dependência entre variáveis através

de cópulas bivariadas em uma estrutura encadeada. Utilizando expansões multivariadas mais flexíveis, através de modelos arquimedianos aninhados, estão os trabalhos de YE *et al* (2010) e CUESTA-INFANTE *et al.* (2010). Este último baseia-se em cópulas empíricas para as distribuições marginais, realizando expansão multivariada de cópulas bivariadas, de forma semelhante à do modelo arquimediano completamente aninhado, porém com parâmetro constante.

A classe dos EDAs baseados em cópulas construídas por meio de modelos *pair-copulas* com vines, os Vine EDAs (VEDAs), é introduzida em SOTO & GONZÁLEZ-FERNÁNDEZ (2010) e GONZÁLEZ-FERNÁNDEZ (2011). Estes autores utilizam vines canônicos, os C-Vine EDAs (CVEDA), e D-Vine EDAs (DVEDA) que também são empregados em SALINAS-GUTIÉRREZ *et al.* (2010). Ambos os CVEDA e DVEDA são implementados no pacote R com as cópulas elípticas normal e t-Student, e as arquimedianas de Clayton, Frank e Gumbel. Nas mesmas, os parâmetros das cópulas são estimados através da inversão do tau de Kendall. Os resultados de simulações realizadas com esses algoritmos foram comparados, no capítulo 5, com os do modelo apresentado nesta tese. O referido modelo será detalhado no capítulo seguinte.

Nos anos de 2013 e 2014, poucos CEDAs foram produzidos. Além dos citados anteriormente, identificam-se na literatura: um algoritmo híbrido de Evolução Diferencial (STORN & PRICE, 1997) com EDA baseado em cópulas empíricas para problemas de otimização dinâmica (SONG & TANG, 2013); um EDA baseado em cópulas Gaussianas com um modelo de migração de parâmetros da cópula e das distribuições marginais – ao invés de migração de indivíduos – entre ilhas de população e aplicado em *benchmarks* de otimização numérica (HYRŠ & SCHWARZ, 2014); EDAs baseados em modelos gráficos em árvore ou em cadeia com flexibilidade na seleção de cópulas para modelagem de dependências bivariadas e aplicados em *benchmarks* de otimização contínua (SALINAS-GUTIÉRREZ *et al.*, 2014); EDAs baseados em cópulas arquimedianas multivariadas com estimação de parâmetros por meio do tau de Kendall da população e aplicados em problemas multiobjetivos (GAO *et al.*, 2014).

4

CEDA aplicado em problemas de otimização numérica (EDA-MEC)

Os capítulos anteriores mostraram que problemas de otimização estão presentes em aplicações de todas as áreas do conhecimento, demandando soluções precisas e com pouco esforço (tempo) computacional. Atender satisfatoriamente a estes requisitos conflitantes é o maior desafio de todos os métodos, sejam os mesmos determinísticos ou estocásticos. Estes últimos encerram, entre outras, a classe dos algoritmos evolutivos, incluindo os algoritmos de estimação de distribuição (EDAs), cujo desempenho está diretamente relacionado à qualidade do modelo probabilístico estimado. Entre as diversas técnicas disponíveis, a estimação por cópulas tem sido recentemente utilizada em EDAs.

O presente capítulo propõe um algoritmo de estimação de distribuição baseado em extensão multivariada de cópulas (*Estimation of Distribution Algorithm based on Multivariate Extension of Copulas*), com acrônimo EDA-MEC. Fazendo uso de técnicas características da computação evolutiva, adicionalmente ao modo distinto com que constrói e utiliza cópulas, o EDA-MEC se mostrou mais eficiente que outros EDAs.

As próximas seções deste capítulo apresentam o algoritmo completo do modelo proposto neste trabalho, destacando os aspectos em que o mesmo difere de outros EDAs baseados em cópulas.

4.1. Algoritmos de estimação de distribuição

4.1.1. Algoritmo de estimação de distribuição genérico

Antes de apresentar as características do EDA-MEC, faz-se oportuno observar o pseudocódigo de um algoritmo de estimação de distribuição genérico, na figura 4.1, muito embora seu fluxograma já tenha sido apresentado na figura 2.4,

na qual foram destacadas as diferenças deste para outro algoritmo evolutivo importante, o algoritmo genético clássico.

EDAs realizam operações, em um instante $t=0$, com uma população $P(t)$ de soluções candidatas (indivíduos) para um problema, inicializada através da amostragem aleatória de uma distribuição uniforme. Estas são argumentos de entrada de uma função objetivo $\Phi(\cdot)$ que avalia a precisão de cada uma delas na solução do problema, sendo ordenadas em uma lista decrescente (*rank*), da melhor para a pior. Baseado nesta avaliação, um subconjunto de soluções é selecionado $S(P(t))$. Então, uma distribuição de probabilidade (modelo) ε é estimada/construída destas soluções e, posteriormente, um conjunto de novas soluções é amostrado do modelo. A nova população $P(t+1)$ pode ser amostrada integralmente da distribuição de probabilidade; ou, visando maior exploração do ambiente, uma porcentagem de indivíduos gerados aleatoriamente pode ser adicionada à nova população; ou ainda apenas alguns indivíduos da população anterior podem ser substituídos pelos novos indivíduos amostrados da distribuição. O processo se repete até que o algoritmo encontre a solução ótima ou que alguma condição de parada seja satisfeita.

Algoritmo 1: EDA genérico

1. Início
 2. $t \rightarrow 0$;
 3. Geração da população inicial $P(t)$ aleatoriamente
 4. **enquanto** (condição de parada não satisfeita) **faça**
 5. Avaliar: $\Phi(P(t))$;
 6. Selecionar: $S(P(t))$;
 7. Estimar distribuição de probabilidade, ε , de $S(P(t))$;
 8. Amostrar uma nova população, $P(t+1)$, de ε ;
 9. $t \rightarrow t + 1$;
 10. **fim**
 11. Fim
-

Figura 4.1 Pseudocódigo de um EDA genérico

4.1.2. Algoritmos de estimação de distribuição baseado em cópulas (CEDAs)

A principal motivação para o uso de cópulas em EDAs está na flexibilidade que as mesmas têm em construir distribuições conjuntas representativas do espaço de busca, em detrimento da suposição comum e conveniente, feita na maioria dos EDAs, que as variáveis seguem uma distribuição gaussiana. Esta conjectura leva frequentemente a resultados inconsistentes e, por conseguinte, a mau desempenho do algoritmo, em especial, nos problemas de otimização numérica em espaços multivariados. Deste modo, a utilização de cópulas em EDAs pode simplificar a operação destes, na medida em que a construção dos modelos probabilísticos é realizada em menor tempo – visto que a distribuição conjunta é obtida das distribuições univariadas – e com maior precisão que outras técnicas de estimação.

De acordo com o teorema de Sklar, apresentado no capítulo anterior, uma função multivariada pode ser representada por uma cópula em conjunto com as distribuições marginais de cada variável. Uma vez que a estrutura de dependência (relação entre as variáveis) é determinada por uma cópula e que a distribuição multivariada F é obtida pela relação (3-1), então a estimação do modelo probabilístico de um EDA baseado em cópula, conforme ilustrado na figura 4.2, consiste basicamente em duas tarefas: i) selecionar ou construir a cópula C ; ii) estimar cada uma das marginais univariadas, F_i , para $i = 1, \dots, n$.

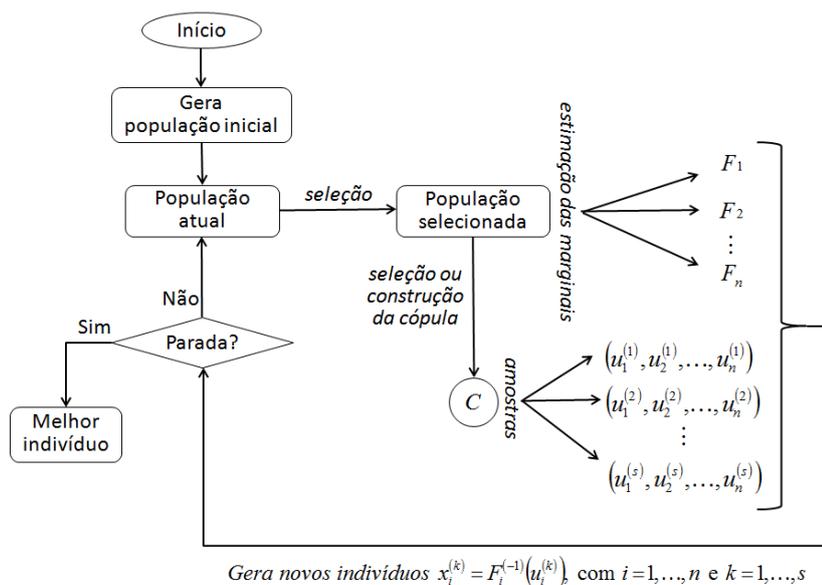


Figura 4.2 Fluxograma de um EDA baseado em cópulas genérico. Adaptado de GUO *et al.* (2011).

Após estimar o modelo de distribuição de probabilidades, realizado com a conclusão das duas tarefas anteriores, o próximo passo consiste em gerar novos indivíduos a partir da amostragem do referido modelo, garantindo que os mesmos seguirão a distribuição estimada. Para gerar um novo indivíduo, o algoritmo gera um vetor $\{u_1, u_2, \dots, u_n\} \in [0,1]^n$ que obedece a C (propriedade básica da teoria das cópulas, apresentada na definição 3.1). Em seguida, os valores x_i são calculados através da função inversa de suas marginais correspondentes, isto é, $x_i = F^{-1}(u_i)$. Então, o novo indivíduo gerado, $\{x_1, x_2, \dots, x_n\}$, é uma amostra que obedece à distribuição multivariada F . Esta sequência de passos de um EDA baseado em cópulas é mostrada no fluxograma da figura 4.2 e, com mais detalhes, no pseudocódigo da figura 4.3.

Algoritmo 2: EDA baseado em cópulas

1. Início
 2. $t \rightarrow 0$;
 3. Geração da população inicial $P(t)$ aleatoriamente
 4. **enquanto** (condição de parada não atingida) **faça**
 5. Avaliar: $\Phi(P(t))$;
 6. Selecionar: m amostras de $P(t)$;
 7. Estimar as marginais. Os m indivíduos são m amostras do vetor aleatório n -dimensional $P(t)$. Cada marginal unidimensional $F_i (i = 1, \dots, n)$ é estimada de acordo com as m amostras. As marginais podem ser distribuições empíricas, normais ou outras;
 8. Simultaneamente ao passo anterior, amostrar indivíduos da cópula C .
Gerar s vetores $\{u_1^{(j)}, u_2^{(j)}, \dots, u_n^{(j)}\}$, com $j = 1, \dots, s$ que obedecem a C , onde a quantidade “ s ” pode ser diferente de “ m ”
 9. Compor a próxima geração $P(t + 1)$ com 3 partes: 1) Manter os k melhores indivíduos da geração atual na próxima geração; 2) Obter s novos indivíduos $x_i^{(j)} = F_i^{-1}(u_i^{(j)})$ para $i = 1, \dots, n$ e $j = 1, \dots, s$. 3) Gerar alguns indivíduos aleatórios no espaço de busca com uma certa taxa de mutação.
 10. $t \rightarrow t + 1$;
 11. **Fim**
 12. Saída: melhor indivíduo da população
 13. Fim
-

Figura 4.3 Pseudocódigo de um EDA baseado em cópulas genérico.

4.2. Modelo proposto

Em referência à seção anterior, as diferenças mais significativas entre EDAs baseados em cópulas residem na:

- Seleção ou construção da cópula;
- Estimação do(s) parâmetro(s) da cópula;
- Estimação das distribuições marginais;
- Amostragem da cópula.

Estes itens constituem o detalhamento das etapas de aprendizagem e construção do modelo probabilístico do EDA, exatamente o gargalo do algoritmo, no qual buscam-se, mesmo com cópulas, a construção de distribuições mais precisas e em menor tempo. Desse modo, justificam-se os esforços na proposição de abordagens mais eficientes relacionadas a estes aspectos do modelo.

As subseções seguintes detalharão cada um dos itens supracitados para o modelo proposto neste trabalho. Antes de apresentar o pseudocódigo do algoritmo do EDA-MEC, uma subseção discutirá técnicas de inserção de diversidade, incluindo os rebeldes e a heurística adaptativa de reinicialização utilizada neste modelo.

4.2.1. Seleção ou construção da cópula

O primeiro passo do EDA-MEC consiste na definição do tipo de cópula a ser utilizada na otimização. No estágio atual de desenvolvimento do código, o usuário pode escolher entre quatro:

- Arquimediana de Clayton;
- Arquimediana de Gumbel;
- Elíptica Gaussiana;
- Elíptica t-Student

A motivação de implementar essas quatro cópulas reside em duas justificativas:

- i) tratabilidade – com expressões analíticas explícitas e fechadas, são as mais simples, inclusive na determinação dos parâmetros;

- ii) aplicabilidade – têm sido as principais utilizadas na modelagem de dados de outras áreas, incluindo a área de hidrologia (REDDY & GANGULI, 2012) e área financeira (CHERUBINI *et al.*, 2004);

Ademais, no caso das arquimedianas, estas possuem a capacidade de representar uma grande variedade de estruturas de dependência, incluindo assimetria e dependência de valores extremos, contornando as principais limitações das cópulas elípticas.

Uma vez que os estudos de casos desta tese consideram funções *benchmarks* com 10, 50 e 100 variáveis, é necessário realizar expansão multivariada das cópulas supracitadas, pois as mesmas são definidas para o caso bivariado. Na prática, a expansão ocorrerá no momento da amostragem, utilizando os algoritmos que serão descritos adiante, na subseção 4.2.4. Os referidos algoritmos precisarão também dos parâmetros das cópulas e das marginais, cujas metodologias de obtenção são descritas nas subseções seguintes.

4.2.2. Estimação das distribuições marginais

De acordo com o teorema de Sklar, expresso nas relações (3-1) e (3-2), se as distribuições marginais F_1, F_2, \dots, F_n fossem conhecidas, amostras da cópula C seriam obtidas de $u_i = F_i(x_i)$ e, por conseguinte, o modelo probabilístico estaria determinado. Todavia, na prática, as marginais são desconhecidas, precisando ser estimadas paramétrica ou não parametricamente.

Os ensaios realizados com EDA-MEC utilizaram dois tipos de marginais, uma paramétrica: normal, e uma não paramétrica: empírica. Esta última considera *pseudo-observações*, uma vez que esta abordagem não faz qualquer suposição acerca da estrutura de dependência das variáveis, diferentemente da primeira.

Na prática, estimam-se as inversas das distribuições marginais e não as distribuições marginais, propriamente ditas. Isto é rigorosamente a aplicação do teorema de Sklar na forma inversa (vide equação 3-2), que combinado com a definição direta (vide equação 3-1) permite entender que:

$$x_i = F_i^{-1}(u_i), \quad (4-1)$$

onde u_i é uma amostra da cópula C_F pertinente à distribuição conjunta F das variáveis aleatórias X_1, X_2, \dots, X_n e $F^{-1}(u) = \inf\{x \in \mathbb{R} \mid F_i(x_i) \geq u\}$ é a inversa

generalizada da distribuição marginal. Esse processo se aplica a qualquer tipo de distribuição marginal.

Lembrando que a função de distribuição cumulativa de uma variável aleatória é não-decrescente e contínua à direita, a inversa generalizada de uma função de distribuição F , $F^{-1}(u)$, é, então, a função quantil de F . Desse modo, está determinada a inversa da marginal empírica. No caso da inversa da marginal normal, utiliza-se a função quantil de uma distribuição normal padrão, isto é, com média zero e desvio padrão 1.

4.2.3. Estimação do(s) parâmetro(s) da cópula

Conforme visto na seção 3.3, o parâmetro das cópulas Gaussiana e t-Student multivariadas é a matriz de correlação linear R da população. Adicionalmente, a cópula t-Student requer a estimação de outro parâmetro: os ν graus de liberdade. Este parâmetro controla a curtose da distribuição e, quando aumenta, torna a distribuição t-Student próxima de uma distribuição normal padrão. Em geral, são aplicados estimadores de máxima verossimilhança (MLE – *Maximum Likelihood Estimator*) para determinar este parâmetro. No caso do EDA-MEC, para reduzir o custo computacional do MLE, quando o usuário escolhe a cópula t-Student, o número de graus de liberdade é fixo e igual à quantidade de novos indivíduos, ou seja, igual ao tamanho da amostra. Dado que os estudos de casos (capítulo 5) utilizam populações de até 30 indivíduos, tem-se, então, ensaios realizados com distribuições t com baixos valores de graus de liberdade. Em consequência disto, ao longo dos ensaios, a distribuição t nunca converge para uma distribuição normal. Entretanto, com essa parametrização, o EDA-MEC tem menos chance de gerar novos indivíduos nos extremos da distribuição, o que pode repercutir no desempenho do mesmo. Buscando verificar tais implicações, na seção 5.7, são realizadas comparações de desempenho do EDA-MEC com as cópulas Gaussiana e t-Student, considerando parametrizações idênticas, inclusive com graus de liberdade obtidos por MLE.

No caso das cópulas arquimedianas, o EDA-MEC utiliza o método da inversão do tau de Kendall para estimar o parâmetro θ da cópula que expressa o grau de dependência entre pares de variáveis. Para este tipo de cópula, conforme

discutido no capítulo anterior, existe uma expressão analítica, apresentada na tabela 3.2, que simplifica consideravelmente o esforço computacional em comparação ao método mais utilizado em EDAs baseados em cópulas, o MLE. GONZÁLEZ FERNÁNDEZ & SOTO (2014), utilizam o método da inversão do tau de Kendall, em alguns dos seus algoritmos, como aproximação inicial para o MLE. Para os estudos de casos apresentados nesta tese, é importante ressaltar que devido à população possuir mais de duas variáveis, a dependência entre as variáveis é expressa por uma matriz de taus de Kendall. Considerando que a expansão multivariada de cópulas arquimedianas adotada no EDA-MEC é do tipo hierárquica aninhada (vide seção 3.4.2) que utiliza um único valor de θ para representar o grau de dependência entre todas as variáveis, tomou-se a média da matriz de taus de Kendall τ . Fazendo uso das relações da tabela 3.2, reproduzidas a seguir, calculou-se o parâmetro θ para cópula de Clayton e de Gumbel::

$$\theta_{\text{Clayton}} = \frac{2\tau_C}{1-\tau_C} \quad (4-2)$$

$$\theta_{\text{Gumbel}} = \frac{1}{1-\tau_G} \quad (4-3)$$

onde τ_C e τ_G são os valores médios da matriz de taus de Kendall das populações utilizadas no EDA-MEC com as cópulas de Clayton e Gumbel, respectivamente. Embora possa constituir uma aproximação muito ruim em determinadas circunstâncias, a representação das dependências entre todas as variáveis por um único parâmetro mostrou-se uma simplificação de relação custo-benefício compensadora, no caso dos experimentos desta tese apresentados adiante. Outra diferença do EDA-MEC para os demais está na atualização do parâmetro θ a cada iteração do algoritmo. A literatura mostra que a maioria dos EDAs baseados em cópulas mantêm este parâmetro fixo ao longo da evolução. Esse é um diferencial do EDA-MEC relativamente aos outros EDAs baseados em cópulas. Ao atualizar o parâmetro θ , re-estimando o mesmo a cada geração, o EDA-MEC aumenta a confiabilidade da distribuição de probabilidade construída, sendo esta mais representativa do espaço de buscas e, conseqüentemente, consegue gerar novos indivíduos que tendem a fazer o algoritmo convergir mais rapidamente para o ótimo global. Além disso, não há grandes esforços computacionais, uma vez que os outros métodos realizam a estimativa desse parâmetro com o computacionalmente custoso MLE.

4.2.4. Amostragem da cópula

Definidos o tipo de cópula, o parâmetro da mesma e as marginais, parte-se para a etapa de amostragem.

CHERUBINI *et al.* (2004) e MAI & SCHERER (2012) apresentam diversos algoritmos de simulação de variáveis aleatórias de cópulas, detalhados a seguir. Os referidos algoritmos foram empregados na geração das variáveis aleatórias das cópulas, u_1, u_2, \dots, u_n . De posse delas, basta utilizá-las na distribuição marginal inversa, conforme (4-1), para obter os novos indivíduos da população.

4.2.4.1. Simulação de cópulas elípticas

No caso das cópulas elípticas, o processo de geração de variáveis aleatórias u_1, u_2, \dots, u_n é bem simples, aplicando-se fielmente o teorema de Sklar: geram-se amostras de uma distribuição multivariada elíptica X_1, X_2, \dots, X_n e, definindo-se o tipo de distribuição marginal F_1, F_2, \dots, F_n , são obtidas as amostras daquela cópula, isto é, $F_1(x_1), F_2(x_2), \dots, F_n(x_n)$.

No caso da cópula gaussiana, implementam-se os seguintes passos:

- Calcula-se a matriz A : a fatoração de Cholesky⁴ da matriz de correlação linear R , isto é, $R = AA^*$, onde A^* é a transposta conjugada de A ;
- Simulam-se n variáveis aleatórias independentes da distribuição normal padrão $N(0,1)$: $\mathbf{z} = (z_1, z_2, \dots, z_n)'$;
- Faz-se $\mathbf{x} = A\mathbf{z}$;
- Faz-se $u_i = \Phi(x_i)$ com $i = 1, 2, \dots, n$ e onde Φ denota a distribuição normal padrão univariada;
- $(u_1, u_2, \dots, u_n)' = (F_1(x_1), F_2(x_2), \dots, F_n(x_n))'$, onde F_i denota a i -ésima distribuição marginal. Para o EDA-MEC, normal ou empírica.

⁴ Fatoração de Cholesky é a decomposição de uma matriz definida positiva (R , neste caso) no produto de uma matriz triangular inferior (A , neste caso) pela sua transposta conjugada.

Passos semelhantes são adotados para gerar amostras da cópula t-student com v graus de liberdade:

- Calcula-se a matriz A : a fatoração de Cholesky da matriz R ;
- Simulam-se n variáveis aleatórias independentes e identicamente distribuídas de $N(0,1)$: $\mathbf{z} = (z_1, z_2, \dots, z_n)'$;
- Simula-se uma variável da distribuição χ_v^2 independente de \mathbf{z} : s ;
- Faz-se $\mathbf{y} = A\mathbf{z}$;
- Faz-se $\mathbf{x} = \sqrt{v/s} \mathbf{y}$;
- Faz-se $u_i = T_v(x_i)$ com $i = 1, 2, \dots, n$ e onde T_v denota a distribuição t-Student univariada;
- $(u_1, u_2, \dots, u_n)' = (F_1(x_1), F_2(x_2), \dots, F_n(x_n))'$, onde F_i denota a i -ésima distribuição marginal. Para o EDA-MEC, normal ou empírica.

4.2.4.2. Amostragem condicional

Trata-se de um método geral para simular variáveis aleatórias de uma dada cópula através de uma abordagem condicional. Basicamente, consiste em gerar pares de observações de variáveis aleatórias uniformemente distribuídas em $[0, 1]$ cuja distribuição conjunta é uma cópula bivariada C , escolhida pelo usuário e com parâmetros conhecidos. Fazendo uso da definição de cópula condicional (seção 3.1.2) e de função inversa, a simulação por amostragem condicional segue o algoritmo:

- Geram-se duas variáveis aleatórias independentes, u e q , de uma distribuição uniforme $(0, 1)$;
- Faz-se $q = C_u(v|u)$;
- Resolve-se v em q : $v = C_u^{-1}(q|u)$;
- Do passo anterior, obtém-se o valor de v que compõe a amostra (u, v) de interesse.

Quando aplicado em cópulas arquimedianas, este método realiza, na prática, expansão multivariada de cópulas bivariadas, através de uma construção baseada em modelos arquimedianos aninhados, com uma única função geradora e um único parâmetro θ , conforme descrito na seção 3.4.2. No caso da cópula de Clayton,

recorrendo às expressões da geradora $\varphi(t)$ e da cópula com $\theta > 0$ da tabela 3.1, são calculadas, inicialmente, as derivadas da função $\varphi^{-1}(t)$, dada a relação destas com a cópula condicional:

$$\begin{aligned}\varphi^{-1(1)}(t) &= -\frac{1}{\theta}(t+1)^{-\frac{1}{\theta}-1}, \quad \varphi^{-1(2)} = \frac{1}{\theta} \frac{\theta+1}{\theta} (t+1)^{-\frac{1}{\theta}-2}, \dots, \\ \varphi^{-1(k)}(t) &= (-1)^k \frac{(\theta+1)(\theta+2)\dots(\theta+k-1)}{\theta^k} (t+1)^{-\frac{1}{\theta}-k}\end{aligned}\quad (4-4)$$

Aplicando a sequência da amostragem condicional, o algoritmo seguinte gera uma variável aleatória $(u_1, u_2, \dots, u_n)'$ da cópula de Clayton:

- Simulam-se n variáveis aleatórias independentes $(v_1, v_2, \dots, v_n)'$ de $U(0, 1)$
- Faz-se $u_1 = v_1$
- Faz-se $v_2 = C_2(u_2|v_1)$, logo

$$\begin{aligned}v_2 &= \frac{\varphi^{-1(1)}(c_2)}{\varphi^{-1(1)}(c_1)} \text{ com } c_1 = \varphi(u_1) = u_1^{-\theta} - 1 \text{ e } c_2 = \varphi(u_1) + \varphi(u_2) \\ &= u_1^{-\theta} + u_2^{-\theta} - 2\end{aligned}\quad (4-5)$$

Assim,

$$v_2 = \left(\frac{u_1^{-\theta} + u_2^{-\theta} - 2}{u_1^{-\theta}} \right)^{-\frac{1}{\theta}-1}\quad (4-6)$$

Igualando-se (4-5) e (4-6), determina-se a segunda variável do par amostrada da cópula de Clayton:

$$u_2 = \left(v_1^{-\theta} \left(v_2^{-\frac{\theta}{\theta+1}} - 1 \right) + 1 \right)^{-\frac{1}{\theta}}\quad (4-7)$$

Inicia-se novamente o algoritmo de amostragem condicional para gerar mais uma variável, fazendo:

$$v_3 = C_3(u_3|u_1, u_2) = \frac{\varphi^{-1(2)}(c_3)}{\varphi^{-1(2)}(c_2)} = \left(\frac{u_1^{-\theta} + u_2^{-\theta} + u_3^{-\theta} - 2}{u_1^{-\theta} + u_2^{-\theta} - 1} \right)^{-\frac{1}{\theta}-2}\quad (4-8)$$

E resolvendo em u_3 :

$$u_3 = \left((u_1^{-\theta} + u_2^{-\theta} - 1) \left(v_3^{-\frac{\theta}{2\theta+1}} - 1 \right) + 1 \right)^{-\frac{1}{\theta}}\quad (4-9)$$

Generalizando para o caso n -variado, obtém-se a equação recursiva:

$$u_n = \left\{ (u_1^{-\theta} + u_2^{-\theta} + \dots + u_{n-1}^{-\theta} - n + 2) \cdot \left(v_n^{-\frac{\theta}{\theta(1-n)-1}} - 1 \right) + 1 \right\}^{-\frac{1}{\theta}}\quad (4-10)$$

Esta abordagem pode ser aplicada a qualquer cópula. No entanto, somente é eficiente quando a função inversa da cópula condicional possui solução analítica. Caso contrário, tem-se que utilizar um algoritmo numérico adicional, aumentando o custo computacional do procedimento. Na cópula de Gumbel, por exemplo, o referido método não é capaz de produzir uma formulação recursiva, semelhante à (4-10) obtida para cópula- n de Clayton. Desse modo, métodos alternativos precisam ser utilizados, tais como o descrito a seguir.

4.2.4.3. Amostragem de Marshall-Olkin

O método proposto por MARSHALL & OLKIN (1988) para a construção composta de cópulas envolve a transformada de Laplace e sua inversa. Relembrando que a transformada de Laplace de uma variável aleatória positiva γ é dada por:

$$\tau(s) = E_{\gamma}(e^{-s\gamma}) = \int_0^{+\infty} e^{-st} dF_{\gamma}(t) \quad (4-11)$$

onde F_{γ} é a função de distribuição de γ , depreende-se que a distribuição é determinada pelo conhecimento de $\tau(s)$. Neste método, a função inversa τ^{-1} atua como função geradora de uma cópula arquimediana, com cálculo facilitado haja vista transformadas de Laplace terem inversas bem definidas.

MARSHALL & OLKIN (1988) consideraram funções de distribuição multivariadas da forma:

$$F(x_1, x_2, \dots, x_n) = E[K(H_1(x)^{\gamma_1}, H_2(x)^{\gamma_2}, \dots, H_n(x)^{\gamma_n})] \quad (4-12)$$

onde γ_i : variável latente positiva;

X_i : variável aleatória cuja condicional é dada por $H_i(x|\gamma_i)^{\gamma_i} = H_i(x)^{\gamma_i}$;

$H_i(\cdot)$: alguma função de distribuição para $i = 1, 2, \dots, n$;

K : uma CDF com marginais uniformes;

E : valor esperado calculado com $\gamma_1, \gamma_2, \dots, \gamma_n$.

Em um caso especial, considerando todas as variáveis latentes iguais entre si, tal que $\gamma_1 = \gamma_2 = \dots = \gamma_n = \gamma$, e CDFs de marginais independentes, MARSHALL & OLKIN (1988) reescreveram (4-12) como:

$$\begin{aligned} F(x_1, x_2, \dots, x_n) &= E[K(H_1(x)^{\gamma_1} \cdot H_2(x)^{\gamma_2} \cdot \dots \cdot H_n(x)^{\gamma_n})] \\ &= \tau\left(\tau^{-1}(F_1(x_1)) + \tau^{-1}(F_2(x_2)) + \dots + \tau^{-1}(F_n(x_n))\right) \end{aligned} \quad (4-13)$$

onde F_i é a i -ésima CDF marginal da CDF conjunta F , e $\tau(\cdot)$ é a transformada de Laplace de γ .

Para gerar X_1, X_2, \dots, X_n com a distribuição de (4-12), FREES & VALDEZ (1998) propuseram o seguinte algoritmo:

- Gera-se uma variável aleatória (latente) com transformada de Laplace τ : γ ;
- Independentemente do passo anterior, geram-se variáveis independentes de $U(0, 1)$: U_1, U_2, \dots, U_n ;
- Calcula-se $X_k = F_k^{-1}(U_k^*)$ em $k = 1, 2, \dots, n$,

onde

$$U_k^* = \tau\left(-\frac{1}{\gamma} \ln U_k\right) \quad (4-14)$$

Este algoritmo, que pode ser implementado facilmente para altas dimensões, constitui a base para a geração de amostras da maioria das cópulas multivariadas de interesse. A única desvantagem, relativamente à amostragem condicional, é a necessidade de simular uma variável a mais, γ . No caso da cópula de Clayton, esta variável é obtida de uma distribuição Gamma(1, $1/\alpha$); para a de Gumbel, é uma variável de uma distribuição $(1/\alpha)$ -Estável. Para outras cópulas, nem sempre é fácil obter esta variável.

Para os estudos de caso desta tese, os seguintes algoritmos foram utilizados na geração de amostras:

Cópula de Clayton

- Gera-se uma variável aleatória de Gamma(1, $1/\alpha$): γ . Assim, γ tem transformada de Laplace $\tau(s) = (1 + s)^{-\frac{1}{\alpha}}$;
- Independentemente do passo anterior, geram-se n variáveis independentes de $U(0, 1)$: U_1, U_2, \dots, U_n
- Calcula-se $X_k = F_k^{-1}(U_k^*)$ em $k = 1, 2, \dots, n$,

onde

$$U_k^* = \tau\left(-\frac{1}{\gamma} \ln U_k\right)$$

Cópula de Gumbel

- Gera-se uma variável aleatória da distribuição Estável(1, 0, 0) com parâmetro $1/\alpha: \gamma$. Assim, γ tem transformada de Laplace $\tau(s) = \exp\left\{-s^{\frac{1}{\alpha}}\right\}$;
- Independentemente do passo anterior, geram-se n variáveis independentes de $U(0, 1): U_1, U_2, \dots, U_n$.
- Para $k = 1, 2, \dots, n$, calcula-se $X_k = F_k^{-1}(U_k^*)$

onde

$$U_k^* = \tau\left(-\frac{1}{\gamma} \ln U_k\right)$$

4.2.5. Técnicas de diversidade

Embora EDAs tenham se mostrado extremamente eficientes na solução de problemas que são difíceis para outras técnicas evolutivas, este tipo de algoritmo apresenta mau desempenho em alguns casos, tais como, em problemas multimodais, multiobjetivos e não estacionários, notadamente devido à convergência muito rápida em conjunto com a ausência de operadores que possam manter ou reintroduzir diversidade na população. Este fenômeno, conhecido como convergência prematura, aumenta o risco de estagnação do algoritmo em ótimos locais. Diante disto, diversas propostas têm sido feitas, tentando preservar a diversidade de soluções na população. As principais utilizam várias subpopulações ou nichos (MAHFOUD, 1995), cada um contendo soluções de diferentes regiões do espaço de busca durante o processo evolutivo.

BOSMAN *et al.* (2008) analisaram a convergência prematura de EDAs em problemas de otimização numérica e propuseram o método AMS (*Anticipated Medium Shift*) para deslocar as novas soluções amostradas \mathbf{x} , na geração $t + 1$, um percentual δ na direção da média de deslocamento previamente observada, isto é: $\mathbf{x} \rightarrow \mathbf{x} + \delta \hat{\boldsymbol{\mu}}^{desloc}(t)$.

Outros métodos de preservação de diversidade fazem uso da mutação. WALLIN & RYAN (2007) e DELAOSSA *et al.* (2009) propuseram abordagens baseadas em “mutação na amostragem” e as aplicaram em problemas de otimização combinatória, obtendo melhorias de desempenho dos EDAs testados, todos não

baseados em cópulas. Nestes trabalhos, a probabilidade complementar de algumas variáveis do problema é utilizada, de acordo com um parâmetro (probabilidade) que define a chance de um indivíduo ser amostrado. Assim, o método consiste em introduzir o operador de mutação na distribuição de probabilidade, adicionando na população indivíduos com baixa probabilidade de serem amostrados do modelo probabilístico, denominados de indivíduos “rebeldes” por DELAOSSA et al. (2009).

O EDA-MEC, imediatamente antes do incremento da iteração, substitui alguns dos indivíduos com pior avaliação por uma variação da definição original dos “rebeldes”. Estes também são amostrados de uma transformação do modelo probabilístico estimado naquela iteração, mas que não necessariamente é o inverso dele, uma vez que o EDA-MEC constrói uma CDF e DELAOSSA et al. (2009), uma PDF. A transformação realizada pelo EDA-MEC consiste em estimar um novo modelo com o complemento das amostras u_n simuladas para cada cópula, com os algoritmos descritos na seção 4.2.3. Isto é, os indivíduos “rebeldes” do EDA-MEC são obtidos de: $x_i = F_i^{-1}(1 - u_i)$. É importante ressaltar que não necessariamente os rebeldes têm que ser obtidos do complemento das amostras, podendo ser gerados de qualquer outra transformação capaz de explorar outras regiões do espaço de busca.

Adicionalmente, uma heurística de reinicialização adaptativa da população também insere diversidade na população. Durante a evolução, a referida reinicialização ocorre sempre que:

- Nas cópulas arquimedianas:
 - O tau de Kendall $\tau = 1$ e, conseqüentemente, o parâmetro da cópula $\theta = \infty$;
- Nas cópulas elípticas:
 - Algum autovalor da matriz de correlação R não for positivo, inviabilizando a fatoração de Cholesky, isto é, não existe uma matriz A em que $R = A \times A^*$;
- Em ambas as cópulas:
 - Alguma variável assume o mesmo valor em todos os indivíduos, o que faz a matriz de correlação gerar valores numéricos não válidos (NaN – *Not a Number*);

- A norma da diferença entre os valores máximo e mínimo dos indivíduos em cada variável normalizada com respeito à distância euclidiana for menor do que 10^{-4} . Quando isto ocorre, os indivíduos estão muito próximos entre si e provavelmente estagnados.

A reinicialização da população é elitista, mantendo sempre o primeiro indivíduo, isto é, o de melhor avaliação. Os demais são substituídos por uma proporção de indivíduos amostrados de uma distribuição uniforme delimitada:

- a) No intervalo do espaço de busca;
- b) Entre os valores máximo e mínimo da população no momento da reinicialização.

A proporção inicial é de 20% de “a)” e 80% de “b)”. Sempre que o EDA-MEC não conseguir encontrar um indivíduo melhor, no intervalo de duas reinicializações consecutivas, a proporção é alterada em 10%, isto é, “a)” diminui para 10% e “b)” aumenta para 90%. No evento seguinte, 0% de “a)” e 100% de “b)”. Posteriormente, 90% de “a)” e 10% de “b)”, e assim por diante.

4.2.6. Detalhamento do algoritmo do EDA-MEC

A seguir o pseudocódigo, contendo os principais passos do algoritmo proposto neste trabalho:

Algoritmo 3: EDA baseado em expansão multivariada de cópulas – EDA-MEC

1. Início
2. Definições: função *benchmark* (Φ); número de variáveis (n), número de indivíduos (m), número de novos indivíduos (y) e rebeldes (r) e condição de parada;
3. Escolha da cópula do modelo: Clayton, Gumbel Gaussiana, t-Student; e o tipo de marginal: normal ou empírica (F_n^{-1});
4. $t \rightarrow 0$;
5. Gere uma população inicial $P(t)$ aleatoriamente no intervalo de busca: matriz (m indivíduos, n variáveis);
6. Avalie $\Phi(P(t))$;
7. **enquanto** (condição de parada não satisfeita) **faça**
8. Calcule a matriz de correlação τ ou R com os dados de $P(t)$: (n, n);

Algoritmo 3: EDA baseado em expansão multivariada de cópulas – EDA-MEC

9. **se** (alguma das condições descritas no item 4.2.5 ocorrer) **faça**
 10. Reinicialize a população $P(t)$ com a heurística do 4.2.5
 11. Calcule a matriz de correlação com os dados de $P(t)$: (n, n) ;
 12. **Fim**
 13. **se** (a cópula for de Clayton ou de Gumbel) **faça**
 14. Estime o parâmetro θ de acordo com as equações (4-2) ou (4-3)
 15. **Fim**
 16. Simule $u_{y,n}$ amostras da cópula escolhida de acordo com os algoritmos do item 4.2.4;
 17. Determine os novos indivíduos utilizando as amostras do passo anterior em $x_{y,n} = F_n^{-1}(u_{y,n})$;
 18. Determine os rebeldes: do passo anterior, calcula-se $1 - u_{y,n}$, selecionando os r primeiros $(u_{r,n})$ e aplicando-os em $x_{r,n}^{rebel} = F_n^{-1}(u_{r,n})$;
 19. Concatene os $x_{y,n}$ novos indivíduos a $P(t)$. Avalie-os. Reordene-os de acordo com a aptidão, mantendo em $P(t)$ os m melhores indivíduos e elimina-se o restante;
 20. Substitua de $P(t)$ os r piores indivíduos por $x_{r,n}^{rebel}$;
 21. $t \rightarrow t + 1$;
 22. **Fim**
 23. Saída: melhor indivíduo da população
 24. **Fim**
-

Figura 4.4 Pseudocódigo do EDA-MEC.

Nos passos 2 e 3, são feitas as definições iniciais para ensaios do algoritmo: função *benchmark* (Φ), o número de variáveis (n), critérios de parada, função cópula, tipo de marginal (F_n^{-1}), número de indivíduos da população (m), número de novos indivíduos amostrados da cópula (y), e número de rebeldes (r).

No passo 5, o algoritmo inicializa uma população aleatoriamente dentro do domínio da função e procede a primeira avaliação, no passo 6. No passo 7, ocorre a etapa de estimação de parâmetros das cópulas, na qual são calculadas medidas de dependência/concordância dos indivíduos da população, através da matriz de correlação linear para as cópulas elípticas ou matriz de taus de Kendall, para as arquimedianas. Se for identificada condição de convergência prematura, o

algoritmo reinicializa sua população, mantendo apenas o melhor indivíduo, substituindo os demais por uma proporção de indivíduos gerados aleatoriamente no espaço de buscas e nos limites máximo e mínimo do melhor indivíduo, no momento da reinicialização. Saindo dessa condição, se a cópula for arquimediana, no passo 13, o parâmetro θ da cópula, que expressa o grau de dependência das variáveis, é calculado, utilizando para isso o valor médio da matriz de taus de Kendall.

No passo 16, são geradas y amostras da cópula, de acordo com os algoritmos de simulação, do item 4.2.4. Utilizando estas amostras, no passo 17, são gerados os y novos indivíduos. No passo 18, após uma transformação nas amostras geradas no passo 16, são gerados r rebeldes. No passo 19, após concatenar os novos indivíduos com os antigos, os mesmos são avaliados, e uma nova população é constituída com os m melhores. Antes de passar para a nova geração, os r piores indivíduos da nova população são substituídos pelos r rebeldes. Inicia-se uma nova geração, caso um critério de parada não tenha sido satisfeito.

A seguir, serão apresentados e discutidos os estudos de casos.

5 Estudos de casos

Em computação evolutiva, é comum comparar algoritmos através de um conjunto de funções de teste ou *benchmarks*. Todavia, a eficiência de um algoritmo de busca não pode ser aferida pela quantidade de problemas que um resolve melhor que o outro, pois se ambos forem testados com todas as funções possíveis, terão, na média, desempenho semelhante (WOLPERT & MACREADY, 1995). Assim, a avaliação de um algoritmo evolutivo deve observar os tipos de problemas em que o mesmo possui bom desempenho, com a finalidade de caracterizar sua adequação (ORTIZ-BOYER *et al.*, 2005). Em decorrência disto, determinados padrões de *benchmarks*, com propriedades variadas, têm sido utilizados para estudar o desempenho dos algoritmos.

Neste capítulo, após definir as seis funções *benchmarks* de otimização numérica utilizadas em GONZÁLEZ-FERNÁNDEZ & SOTO (2012)⁵ e relacionar suas propriedades, são apresentados os resultados de diferentes ensaios com o algoritmo do modelo proposto. Foram considerados quatro tipos de cópulas, dois tipos de distribuições marginais e variados números de indivíduos e rebeldes, comparando o desempenho do EDA-MEC com outros EDAs baseados em cópulas (GONZÁLEZ-FERNÁNDEZ & SOTO, 2012) e com o CMA-ES.

5.1. Propriedades de funções utilizadas como *benchmarks*

Funções de teste *benchmarks* são problemas artificiais utilizados para avaliar o desempenho de um algoritmo em situações diferentes e difíceis. Estes problemas podem ser facilmente manipulados e modificados para testar algoritmos de otimização em diversos cenários. As seguintes características de *benchmarks* são particularmente desafiadoras para algoritmos de otimização:

⁵ São utilizados os resultados de GONZÁLEZ-FERNÁNDEZ & SOTO (2012), pois a versão mais recente, publicada em 2014, considera um número menor de funções *benchmarks*.

- Não separabilidade;
- Multimodalidade;
- Não convexidade;
- Escalabilidade.

Na literatura de otimização contínua, o conceito de não separabilidade está associado à interação de variáveis. Diz-se que duas ou mais variáveis são não separáveis quando a tarefa de localização do ótimo global de uma função objetivo não pode ser realizada através da otimização independente de uma relativamente às outras. Na computação evolutiva, este fenômeno é conhecido como *epistasia* e mede o nível de interação dos genes de um indivíduo, ou seja, quanto o valor de um gene interfere no valor de outros genes. Funções não separáveis são mais difíceis de otimizar quando a direção exata da busca depende de dois ou mais genes. Por outro lado, funções separáveis podem ser otimizadas em cada variável individualmente.

O problema se torna ainda mais difícil quando a função é multimodal, isto é, quando possui mais de um ótimo local. Estas funções são utilizadas para testar a habilidade de um algoritmo de otimização em sair de quaisquer ótimos locais. Os casos mais complexos ocorrem quando estes estão distribuídos aleatoriamente no espaço de busca.

O conceito de convexidade é também fundamental em otimização. Se uma função e seu domínio são convexos, então qualquer solução local do problema é de ótimo global. Por outro lado, não existem condições matemáticas que garantam a convergência de funções não convexas para o ótimo global.

Outra forma de aumentar a complexidade do processo de busca é através do aumento do número de variáveis e de restrições das funções *benchmarks*. Primeiramente, EDAs sofrem da chamada maldição da dimensionalidade (DONG *et al.*; FRIEDMAN, 1994): a quantidade de dados necessária para sustentar uma dada densidade espacial cresce exponencialmente com a dimensionalidade do espaço de busca. Em segundo lugar, as propriedades do espaço de busca podem mudar com o aumento do número de dimensões. Por fim, a avaliação de problemas de grande escala requer grande esforço computacional.

5.2. Funções *benchmarks* utilizadas em otimização numérica

Há muitas funções *benchmarks* descritas na literatura; entretanto, não existe uma lista ou conjunto padrão de *benchmarks*. Idealmente, estas funções deveriam possuir diversas propriedades, tais quais as relacionadas no item anterior, para serem efetivamente úteis na avaliação imparcial de algoritmos de otimização.

No caso de EDAs baseados em cópulas, verifica-se uma escassez de publicações com resultados de testes de desempenho disponíveis. Uma destas, bem recente, que inclusive disponibiliza um pacote em linguagem R (copulaedas) com implementação de vários EDAs baseados em cópulas, é GONZÁLEZ-FERNÁNDEZ & SOTO (2012, 2014). Desse modo, as seis funções *benchmarks* e os espaços de buscas correspondentes definidos em GONZÁLEZ-FERNÁNDEZ & SOTO (2012) são os mesmos utilizados na avaliação do EDA-MEC, conforme descrição a seguir:

- Função esfera ou primeira função de De Jong

Trata-se de uma das funções mais simples utilizadas em testes de algoritmos evolutivos. A mesma é contínua, convexa, unimodal, separável e pode ser ampliada para qualquer número de variáveis. É uma função do tipo quadrática, possuindo um único ótimo local (e, portanto, global) em $x = (0, \dots, 0)$. O espaço de busca é definido em $[-600, 600]$ para cada variável de decisão. Sua forma analítica é dada por:

$$f_{sphere}(x) = \sum_{i=1}^n x_i^2, \quad (5-1)$$

onde x é um vetor de decisão de n variáveis ou dimensões. Sua representação para o caso de duas variáveis – reduzida para o intervalo de $[-100, 100]$ com a finalidade de melhorar a visualização de suas características – é dada na figura 5.1.

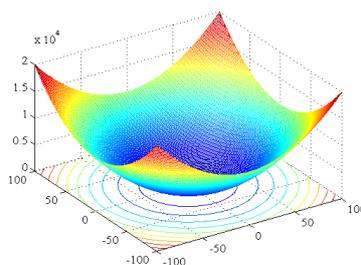


Figura 5.1 - Mapa 3D da função esfera de 2 variáveis, $f(x, y) = x^2 + y^2$.

- Função de Ackley

Trata-se de uma função multimodal, separável e com vários ótimos locais que no domínio $[-30, 30]$ mais parecem ruído, embora estejam localizados em intervalos regulares. Possuindo ótimo global em $x = (0, \dots, 0)$, a função de Ackley constitui um problema de complexidade moderada e sua forma analítica é dada por:

$$f_{Ackley}(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \exp(1), \quad (5-2)$$

onde \mathbf{x} é um vetor de decisão de n variáveis ou dimensões. Sua representação para o caso de duas variáveis – reduzida para o intervalo de $[-5, 5]$ com a finalidade de melhorar a visualização de suas características – é dada na figura 5.2.

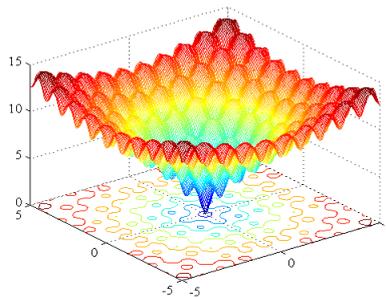


Figura 5.2 - Mapa 3D da função de Ackley de 2 variáveis.

- Função de Rastrigin

É uma função não convexa, multimodal e separável. Possui vários ótimos locais arranjados em um grafo regular (*regular lattice*), com o ótimo global localizado em $x = (0, \dots, 0)$. O espaço de busca é definido no intervalo $[-5, 12; 5, 12]$ em cada variável. Constitui um problema razoavelmente difícil devido à quantidade grande de ótimos locais. Sua forma analítica é dada por:

$$f_{Rastrigin}(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad (5-3)$$

onde \mathbf{x} é um vetor de decisão de n variáveis ou dimensões. Sua representação para o caso de duas variáveis é dada na figura seguinte.

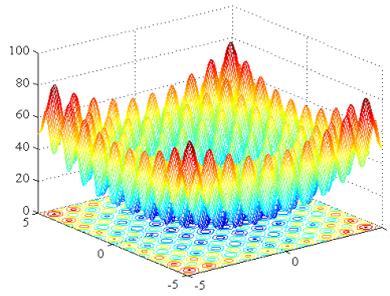


Figura 5.3 - Mapa 3D da função de Rastrigin de 2 variáveis.

- Função de Griewank

Esta função é multimodal e não separável, com vários ótimos locais dentro do espaço de busca, definido no intervalo $[-600, 600]$. É semelhante à função de Rastrigin, mas a de Griewank possui maior número de ótimos locais. A mesma possui ótimo global localizado em $x = (0, \dots, 0)$. A interpretação desta função se modifica com a escala; a visão geral sugere uma função convexa, a visão de média escala sugere a existência de mínimos locais e, por fim, o exame mais próximo permite identificar uma estrutura complexa de inúmeros mínimos locais. Verifica-se na estrutura analítica, relacionada a seguir, que um termo de produto introduz interdependência entre as variáveis. O objetivo disto é fazer com que as técnicas, que otimizam cada variável independentemente, falhem.

$$f_{Griewank}(\mathbf{x}) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad (5-4)$$

onde \mathbf{x} é um vetor de decisão de n variáveis ou dimensões. Sua representação para o caso de duas variáveis – reduzida para o intervalo de $[-10, 10]$ com a finalidade de melhorar a visualização de suas características – é dada na figura 5.4. Neste caso, tem-se a visão descrita anteriormente de inúmeros mínimos locais.

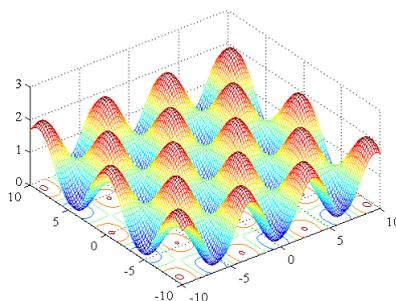


Figura 5.4 - Mapa 3D da função de Griewank de 2 variáveis.

- Função Summation Cancellation

Esta tese utiliza a mesma definição de GONZÁLEZ-FERNÁNDEZ & SOTO (2012), a qual modifica o problema original (BALUJA & CARUANA, 1995) de maximização em um de minimização. Esta função, de acordo com a equação seguinte, possui interações lineares multivariadas crescentes em y_i , uma vez que a variável y_i é definida em termos de todas as y_j variáveis, com $j < i$. Logo, nesta função, algoritmos capazes de modelar dependências lineares devem superar os que não são. Além disso, devido ao fato do ótimo global estar situado em um pico bem acentuado e estreito próximo a ótimos locais, o algoritmo precisa ter grande precisão e habilidade para evitar convergência prematura. O valor mínimo desta função é de $-1E5$ e ocorre em $x = (0, \dots, 0)$. E o espaço de busca está definido no intervalo $[-0.16, 0.16]$.

$$f_{summ\ cancell}(x) = \frac{-1}{10^{-5} + \sum_{i=1}^n |y_i|}, y_1 = x_1, y_i = y_{i-1} + x_i \quad (5-5)$$

Sua representação para o caso de duas variáveis – reduzida para o intervalo de $[-0,02, 0,02]$ com a finalidade de melhorar a visualização de suas características – é dada na figura 5.5.

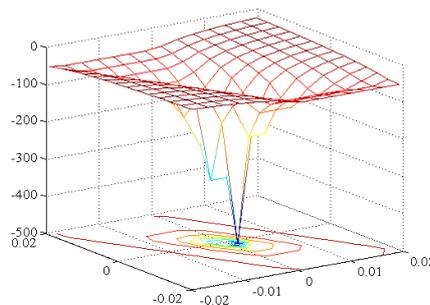


Figura 5.5 - Mapa 3D da função modificada Summation Cancellation de 2 variáveis.

- Função de Rosenbrock ou segunda função de De Jong

Também conhecida como vale ou função banana de Rosenbrock, consiste em uma função não convexa, unimodal e não separável. O mínimo global está dentro de um vale plano, longo, estreito e em formato parabólico. Os algoritmos de otimização chegam ao vale com facilidade, mas têm dificuldades de convergir para o mínimo global, localizado em $x = (1, \dots, 1)$. Esta função também é quadrática e tem seu domínio definido no intervalo de $[-9, 11]$. Sua forma analítica é dada por:

$$f_{Rosenbrock} = \sum_{i=1}^{n-1} (100(x_i^2 - x_{1+i})^2 + (x_i - 1)^2), \quad (5-6)$$

onde x é um vetor de decisão de n variáveis ou dimensões. Sua representação para o caso de duas variáveis – reduzida para o intervalo de $[-2, 3]$ com a finalidade de melhorar a visualização de suas características – é dada na figura 5.6.

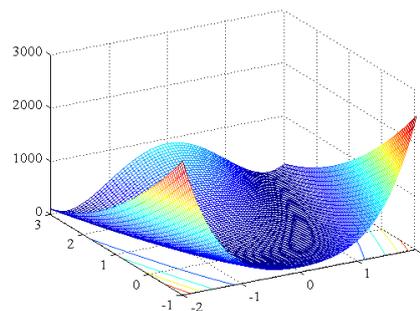


Figura 5.6 - Mapa 3D da função de Rosenbrock de 2 variáveis.

5.3. Metodologia geral dos testes

O EDA-MEC foi testado com as seis funções *benchmarks* apresentadas na seção anterior. A fim de estabelecer comparações com os resultados de GONZÁLEZ-FERNANDEZ & SOTO (2012), condições semelhantes deste foram adotadas nos ensaios:

- Todas as funções foram simuladas com 10 variáveis, isto é, $n = 10$, com os domínios correspondentes definidos na seção anterior;
- Cada ensaio envolveu a realização de 30 experimentos independentes por função, sendo calculados, ao final, a média e o desvio padrão das aptidões (valores mínimos obtidos nas funções) e do número de avaliações; e identificados a melhor e a pior aptidão da última avaliação dos 30

experimentos, isto é, da última iteração do algoritmo na qual são calculadas a média e desvio padrão;

- Foram estipuladas duas condições de parada do algoritmo: i) alcançar o mínimo global da função com precisão maior que $1E-6$; ou ii) atingir o número máximo de 300.000 avaliações. Estes critérios diferem dos definidos na sessão especial de otimização real do Congresso de Computação Evolutiva de 2005 (CEC'05), apesar das funções *benchmarks* serem similares às da mesma (SUGANTHAN *et al.*, 2005).

Os ensaios realizados por GONZÁLEZ-FERNANDEZ & SOTO (2012) consideraram ainda um terceiro critério de parada: iii) perda de diversidade, quando o desvio padrão da avaliação das soluções é menor que $1E-8$. O EDA-MEC não adotou este critério visto que procede reinicializações da população durante o processo de busca, nas situações descritas na seção 4.2.5. Aliás, um dos melhores algoritmos evolutivos em problemas de otimização numérica, o CMA-ES também promove reinicializações na população, alternando melhoras e pioras nas soluções encontradas durante o processo de busca. No caso do EDA-MEC, a reinicialização da população não descarta o melhor indivíduo e, por isso, não existem momentos de piora na evolução. Além disso, o EDA-MEC não aumenta o número de indivíduos após as reinicializações, diferentemente do CMA-ES.

Um parâmetro fundamental em algoritmos evolutivos é a quantidade de indivíduos da população. HAUSCHILD & PELIKAN (2011) discutem que a mesma não pode ser pequena, a ponto de prejudicar a exploração do espaço de busca, e nem grande demais, situação em que aumenta a complexidade da construção e amostragem do modelo probabilístico. Além disso, a quantidade de indivíduos determina a contagem do número de avaliações de um algoritmo evolutivo. GONZÁLEZ-FERNANDEZ & SOTO (2012) construíram, no pacote R, uma rotina baseada no método da bissecção que determina empiricamente a quantidade mínima de indivíduos (tamanho crítico) exigida pelo algoritmo para encontrar o ótimo global da função com uma taxa alta de sucesso e que deve ser utilizada nos ensaios. O método da bissecção inicia no intervalo de [50, 2000] indivíduos, no qual o tamanho crítico da população deve estar localizado, descartando a metade do intervalo a cada passo.

Antes de proceder comparações com os resultados dos algoritmos implementados em GONZÁLEZ-FERNANDEZ & SOTO (2012) e com o

CMA-ES, alguns ensaios foram realizados para avaliar o impacto no desempenho do EDA-MEC com variação dos seguintes parâmetros:

- Algoritmos de amostragem: condicional e de Marshall-Olkin;
- Distribuições marginais: empírica e normal;
- Número de indivíduos na população: 15, 20 e 30;
- Número de rebeldes: 0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 12 e 18.

A referida parametrização foi avaliada com quatro cópulas: Clayton, Gumbel, Gaussiana e t-Student.

5.4. Testes de parametrização

Os resultados apresentados nas seções seguintes são decorrentes de análises estatísticas que buscam identificar as configurações capazes de produzir o número máximo de convergências para o ótimo global em cada *benchmark* com o menor número de avaliações e de tempo de máquina do algoritmo. Os ensaios foram realizados em máquinas com processador Intel(R) Core(TM) i7-3770 de 3.4 GHz.

5.4.1. Ensaio com algoritmos de amostragem na cópula de Clayton

O desempenho médio do EDA-MEC foi avaliado com os algoritmos de amostragem condicional e de Marshall-Olkin, especificamente para a cópula de Clayton, visto que a abordagem condicional não é eficiente para outras cópulas arquimedianas (seção 4.2.3.2). Adicionalmente, o impacto da escolha das distribuições marginais também foi verificado em termos do tempo médio de máquina e da taxa média percentual de sucesso, conforme as figuras 5.7 e 5.8. É importante ressaltar que a taxa média de 100% ocorre quando o EDA-MEC consegue identificar o ótimo global em todos os 30 experimentos de cada ensaio, considerando os diferentes números de indivíduos de população e rebeldes.

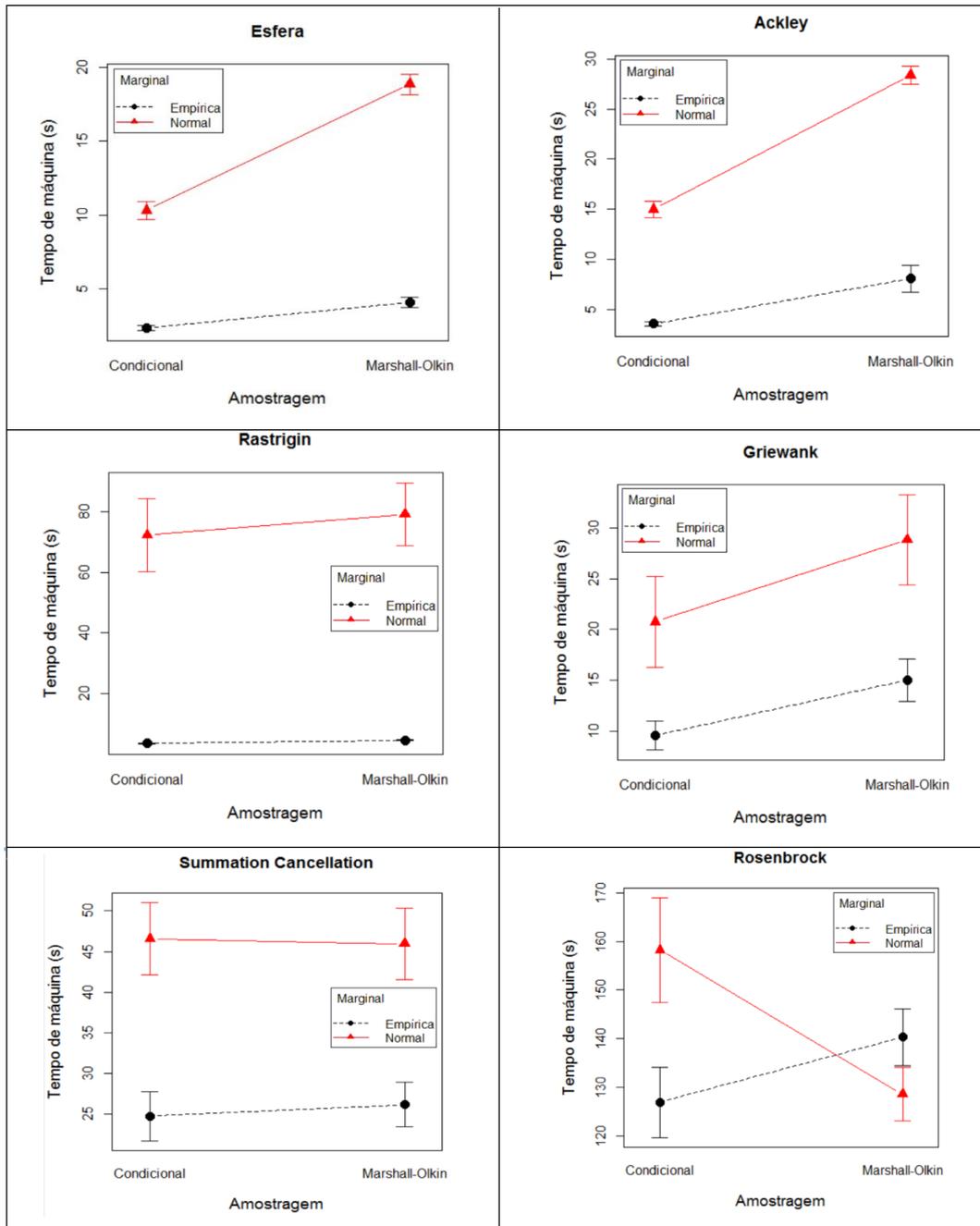


Figura 5.7 - Tempo médio de máquina do EDA-MEC, considerando duas formas de amostragem da cópula de Clayton e duas distribuições marginais.

A análise do tempo de máquina evidencia que, na média, a configuração mais eficiente, de acordo com as funções, ocorre com:

- amostragem condicional;
- distribuições marginais empíricas.

Este resultado é consistente com a teoria, visto que o algoritmo de Marshall-Olkin realiza a simulação de trinta variáveis por geração a mais que o algoritmo de amostragem condicional, demandando, assim, tempo computacional maior. Com

exceção da Summation Cancellation, a diferença de tempo de máquina entre os dois tipos de amostragem é mais evidente ainda quando a modelagem das marginais é feita com distribuição normal.

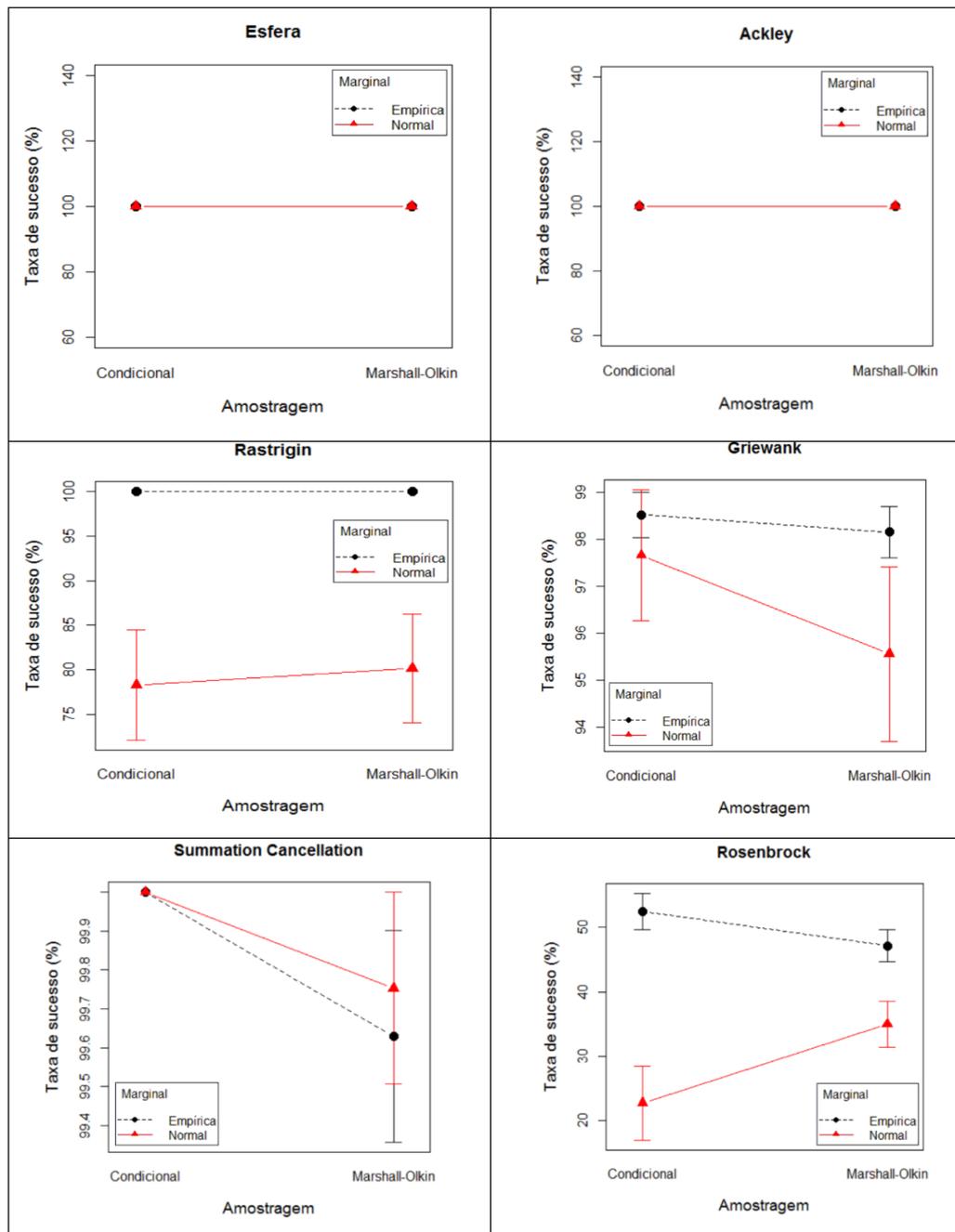


Figura 5.8 - Taxa média percentual de sucesso do EDA-MEC, considerando as amostragens condicional e de Marshall-Olkin da cópula de Clayton e duas distribuições marginais.

A figura 5.8 mostra as configurações que resultam, na média, em maior taxa de sucesso:

Taxa média = 100%

- Esfera e Ackley: qualquer amostragem e qualquer marginal;
- Summation Cancellation: amostragem condicional e qualquer marginal.

Taxa média < 100%

- Rastrigin: qualquer amostragem e marginal empírica;
- Griewank e Rosenbrock: amostragem condicional e marginal empírica;

Assim, a amostragem condicional é um requisito para otimização da Summation Cancellation em 100% dos casos. E, considerando que esse tipo de amostragem também resulta em menor tempo médio de máquina, os próximos ensaios com a cópula de Clayton irão considerar exclusivamente a amostragem condicional.

Apesar da modelagem das marginais com distribuição empírica ter contribuído para melhor desempenho do EDA-MEC, repise-se que este é um resultado médio. Adiante, será mostrado que a configuração que utiliza amostragem condicional e marginal normal resulta em alguns ensaios nos quais a Rosenbrock é otimizada em todos os 30 experimentos, ainda que esta configuração não seja a que gera a melhor média. Em função disto, os próximos ensaios irão considerar os dois modelos de distribuições marginais, empírica e normal, com a cópula de Clayton.

5.4.2. Ensaio com as distribuições marginais

O impacto da escolha das distribuições marginais também foi verificado nas demais cópulas através do número médio de avaliações (*AES – Average number of Evaluation to Solution*) e da taxa percentual de sucesso, apresentados nas figuras 5.9 e 5.10, respectivamente.

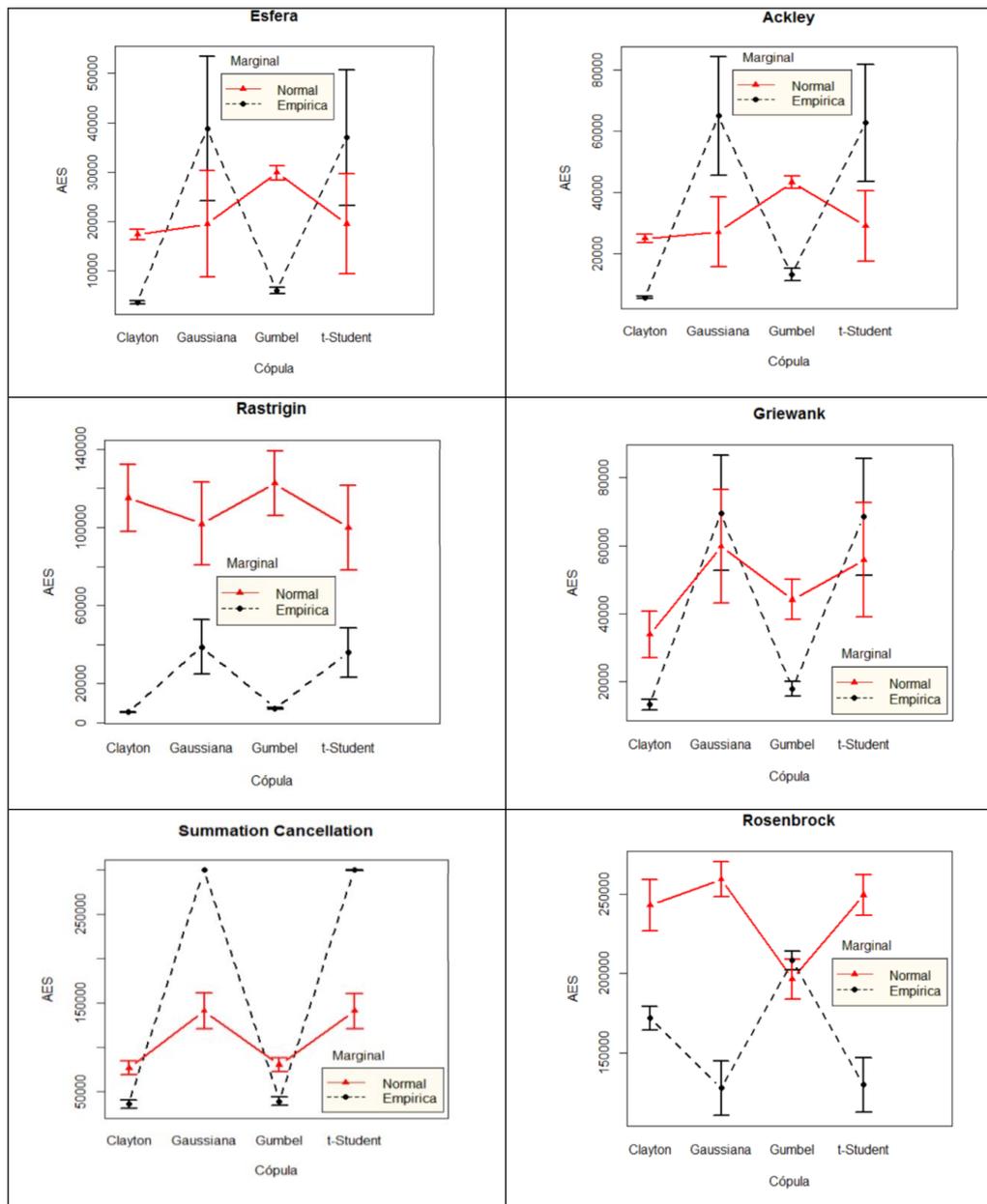


Figura 5.9 - Número médio de avaliações (AES) do EDA-MEC, considerando a interação de quatro tipos de cópulas com duas distribuições marginais.

Confirmando os resultados da subseção anterior, as cópulas arquimedianas, de Clayton e de Gumbel, apresentam, na média, melhor AES com distribuições marginais empíricas em todas as funções. O único caso em que a escolha da marginal normal equipara este critério de desempenho com a marginal empírica ocorre na função de Rosenbrock com a cópula de Gumbel. Por outro lado, as cópulas elípticas, Gaussiana e t-Student, apresentam, na média, melhor desempenho quando suas marginais são normais, nas funções: Esfera, Ackley e Summation. Isto pode ser explicado pela simplicidade das interações de variáveis, nas quais a

modelagem por uma distribuição normal constitui aproximação de bom custo-benefício.

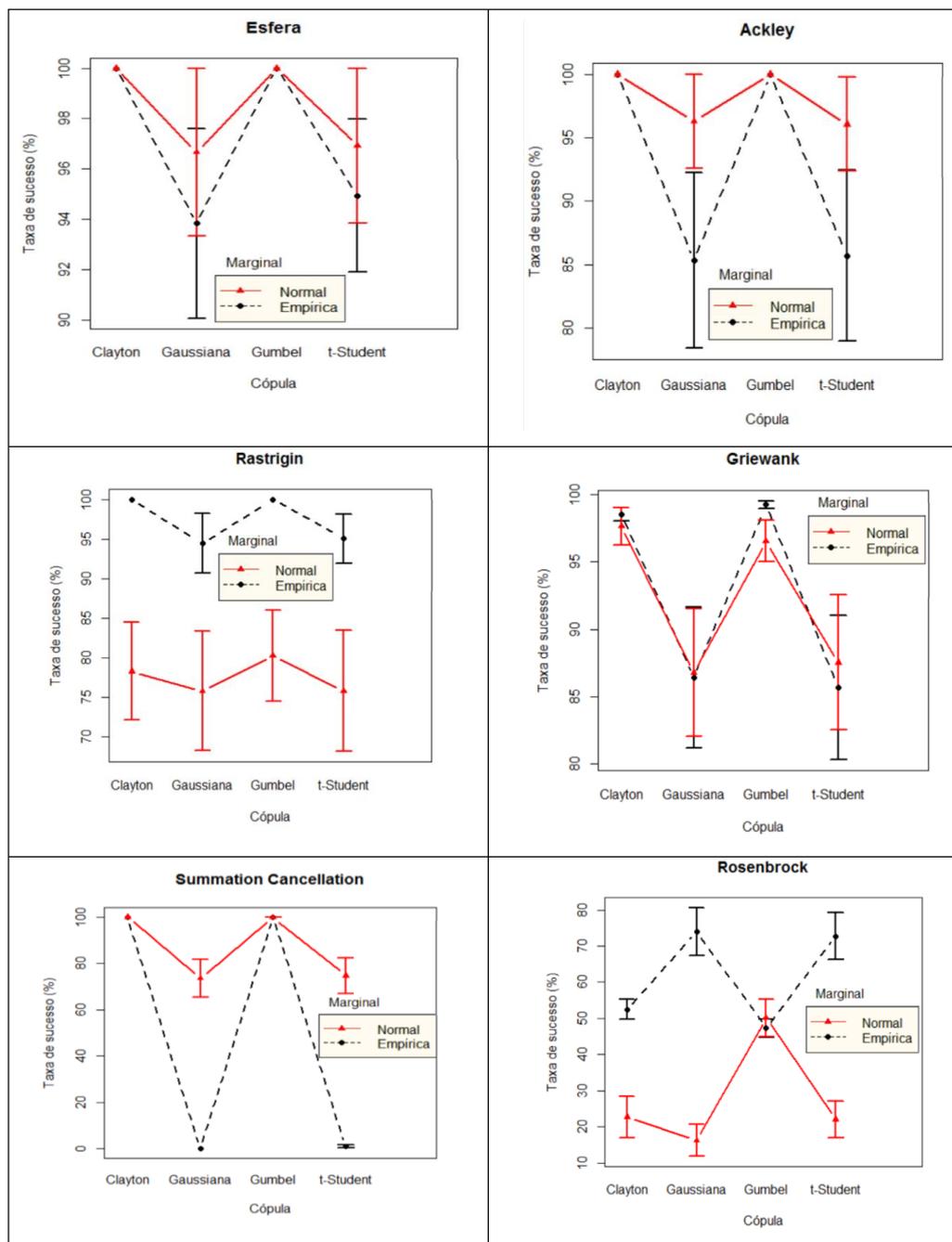


Figura 5.10 - Taxa média percentual de sucesso do EDA-MEC, considerando a interação de quatro tipos de cópulas com duas distribuições marginais.

A eficiência do EDA-MEC em termos de convergência, visualizada na figura 5.10, é percebida novamente, na média, com a utilização de distribuições empíricas, no caso das cópulas arquimedianas. A utilização de marginais normais só produz bons resultados, semelhantes ao das empíricas, na função de Rosenbrock, com a cópula

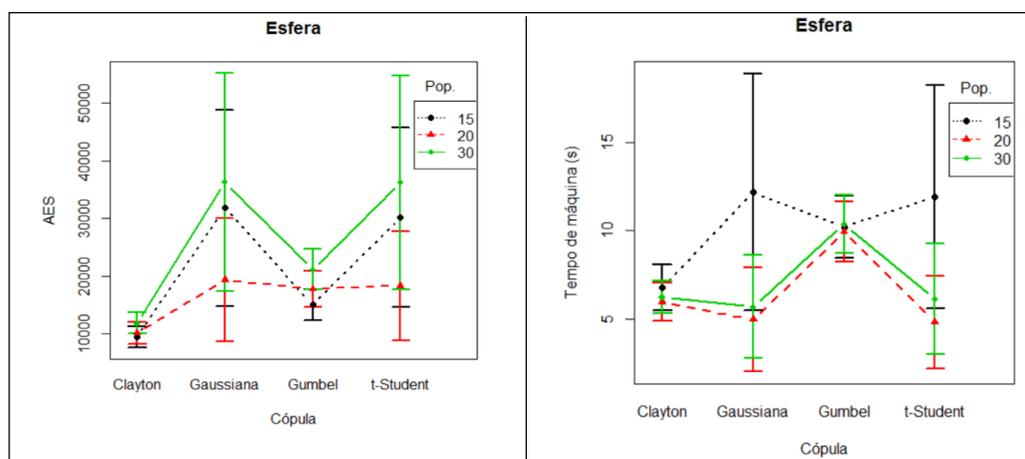
de Gumbel. No caso das cópulas elípticas, verifica-se que o EDA-MEC consegue melhor taxa média de sucesso com marginais normais, nas funções Esfera, Ackley e Summation, comportamento semelhante ao verificado com as AES da figura 5.9. Por se tratar de uma análise sobre a média, não se pode descartar a utilidade das marginais normais nas cópulas arquimedianas e nem das empíricas nas cópulas elípticas.

5.4.3. Ensaio com números diferentes de indivíduos na população

O desempenho do EDA-MEC, em termos de número de avaliações, de tempo de máquina e da taxa percentual de sucesso, não pode ser aferido exclusivamente do número de indivíduos na população definido no início de cada ensaio. Outro fator importante repercute sobre o mesmo: o número de reinicializações. Para ilustrar esta interação, as figuras seguintes mostram a relação entre o tipo de cópula e o número de indivíduos com as médias:

- da AES;
- do tempo de máquina;
- da taxa percentual de sucesso;
- do número de reinicializações.

Os resultados desses ensaios são apresentados somente para as funções Esfera e Griewank porque foi observado o mesmo padrão de comportamento para as demais funções, sendo, portanto, desnecessário reproduzi-los.



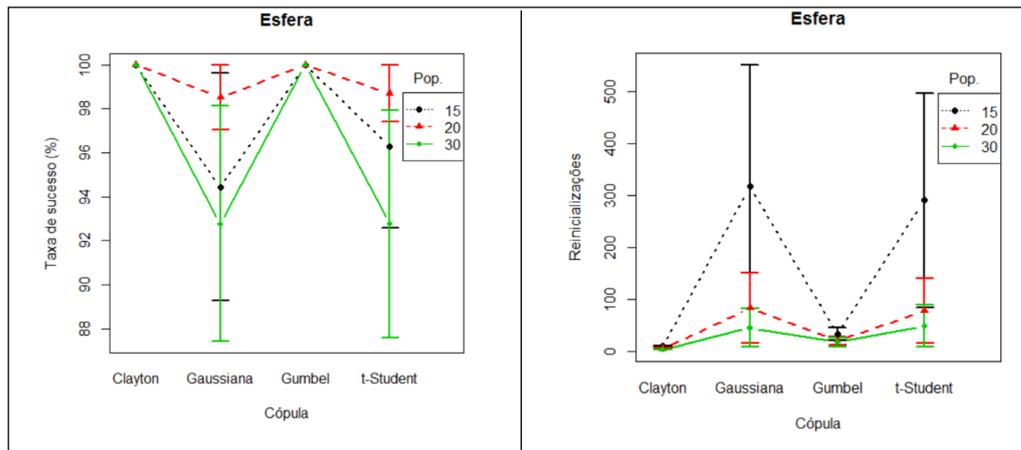


Figura 5.11 - Comportamento de quatro medidas relacionadas ao desempenho do EDA-MEC com a variação da cópula e tamanho da população na função Esfera.

Na figura 5.11, a aparente inconsistência de um maior número médio de avaliações (AES) simultaneamente à baixa média de tempo de máquina para as cópulas elípticas, na população de 30 indivíduos, relativamente ao comportamento da população de 15 indivíduos nestas mesmas cópulas, é esclarecida pelo número médio de reinicializações que ocorrem em cada caso: a população de 15 reinicializa em média 10 vezes mais do que a de 30, demandando com isto mais tempo de máquina. No entanto, esta proporção não se traduz na média de AES, isto é, as reinicializações nas duas populações equiparam a quantidade de indivíduos das mesmas, resultando, portanto, em um número médio de AES muito próximo para elas, visualizado nas curvas do gráfico do primeiro quadrante da figura 5.11. Por fim, neste caso, as reinicializações da população de 15 não repercutem em uma significativa maior taxa média de sucesso comparativamente à de 30. Na função esfera, com os três tamanhos de população, as cópulas arquimedianas auxiliaram o EDA-MEC a obter, na média, melhores resultados que as elípticas.

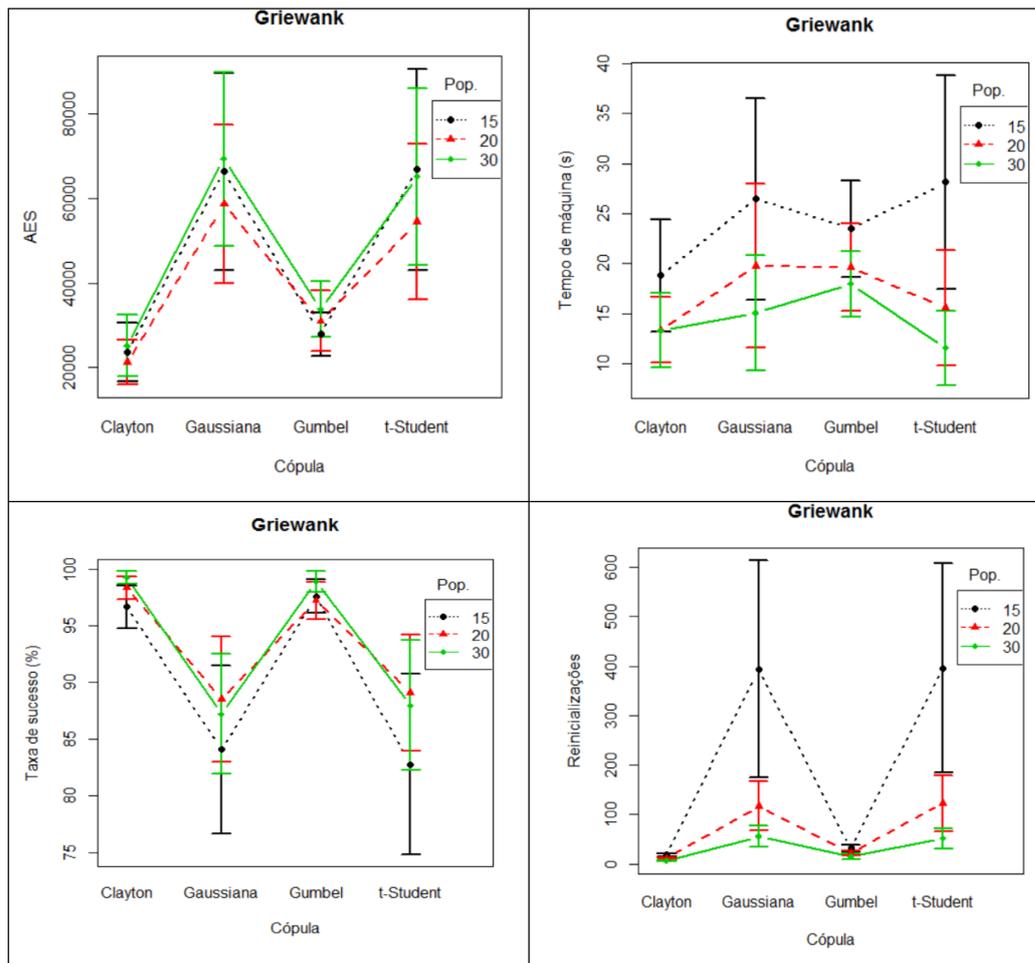


Figura 5.12 - Comportamento de quatro medidas relacionadas ao desempenho do EDA-MEC com a variação da cópula e tamanho da população na função de Griewank.

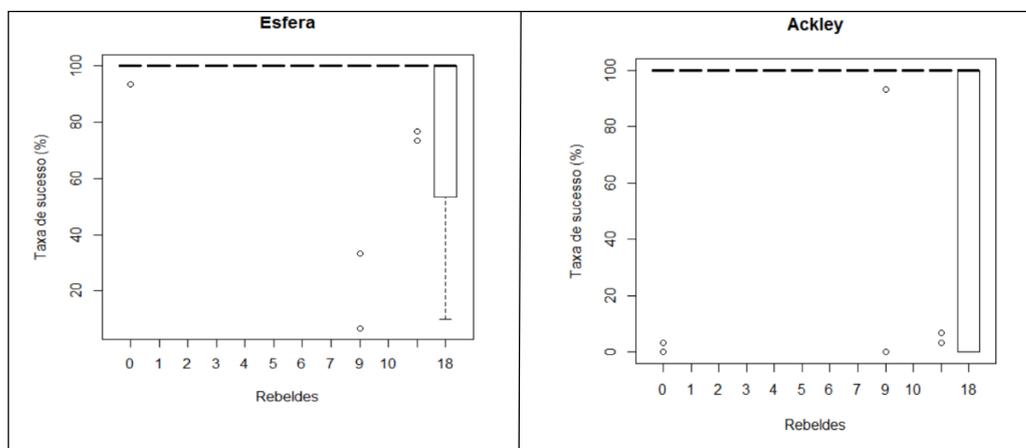
No caso da função de Griewank, verifica-se novamente o que ocorre com todas as demais funções: nas cópulas elípticas, populações menores provocam maior número de reinicializações do EDA-MEC e, conseqüentemente, maior tempo de máquina, o que não resulta em proporcional aumento do número de avaliações (AES), pois, na média, as populações se equiparam em quantidades de indivíduos. Apesar do maior número de reinicializações das populações não contribuir necessariamente para que estas superem as de populações maiores em termos de média de taxa de sucesso do EDA-MEC, as referidas reinicializações cumprem um papel importante ao corrigir a convergência prematura, permitindo que o modelo alcance bons resultados. No caso da função de Griewank, as cópulas arquimedianas permitiram o EDA-MEC obter melhor desempenho nos ensaios com os 3 tamanhos de população, particularmente com 30 indivíduos, em comparação ao obtido com as cópulas elípticas. Repita-se que este é um resultado médio, sendo identificados

ensaios em que as cópulas elípticas obtiveram resultados tão bons quanto as arquimedianas.

É importante ressaltar a importância das reinicializações para o EDA-MEC. Ensaios não apresentados nessa tese mostraram que o algoritmo estagna com a desabilitação da função que promove reinicializações. Complementando as circunstâncias descritas na seção 4.2.5, constatou-se que as reinicializações ocorreram sempre que os indivíduos encontravam-se pouco dispersos, isto é, quando o desvio padrão da população convergia para zero. No entanto, não foi observado qualquer comportamento regular acerca da média e desvio padrão da matriz de taus de Kendall que permitisse generalizar uma circunstância implícita de reinicialização.

5.4.4. Ensaios com número de rebeldes

Conforme destacado no capítulo anterior, a inserção de rebeldes na população, inovação contida no EDA-MEC, auxilia este EDA baseado em cópulas a lidar, em conjunto com as reinicializações, com o problema da convergência prematura. A figura seguinte ilustra a repercussão dos rebeldes no desempenho do EDA-MEC na otimização de cada uma das seis funções *benchmarks* em ensaios com as populações de 15, 20 e 30 indivíduos; com as cópulas Gaussiana, t-Student, Clayton, Gumbel; e com as distribuições marginais empírica e normal. Assim, os desvios padrões da figura 5.13 envolvem o uso de diferentes populações, cópulas e tipos de distribuição marginal.



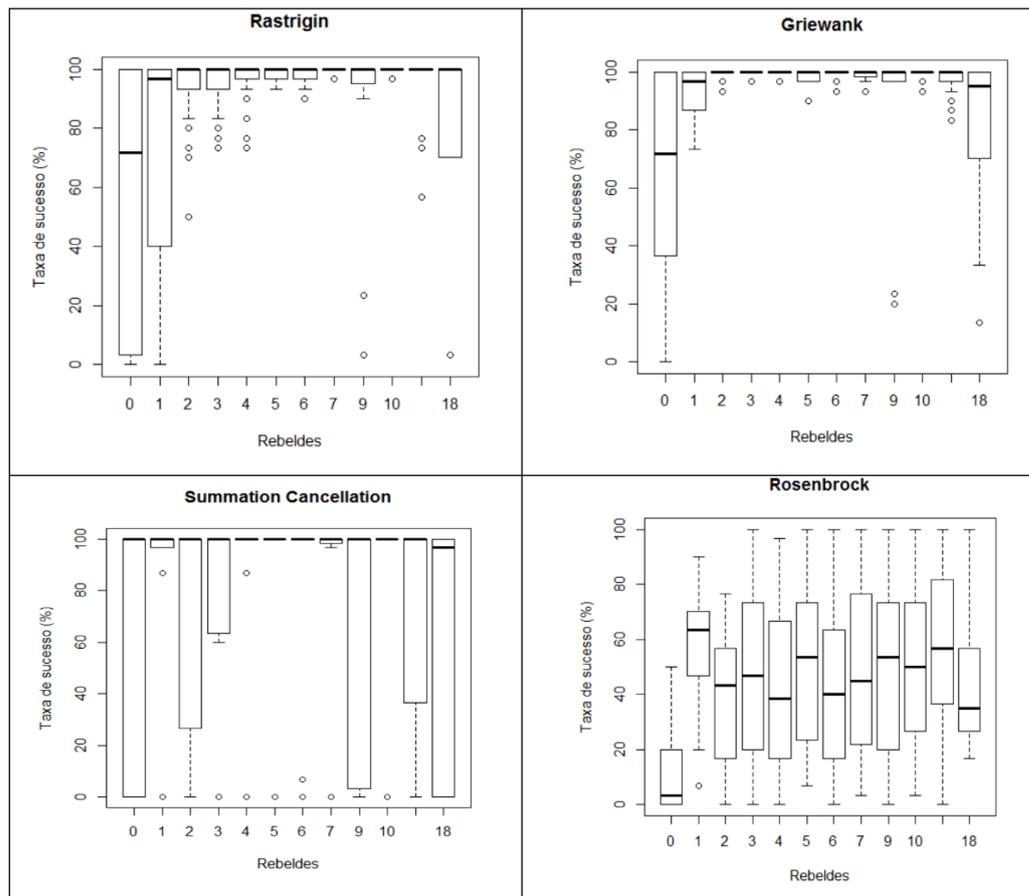


Figura 5.13 - Influência dos rebeldes na convergência do EDA-MEC nas seis funções *benchmarks* considerando todas as configurações de ensaios por função.

A leitura dos gráficos da figura 5.13 permite concluir que as funções mais simples, Esfera, Ackley e Summation, não requerem rebeldes para que o EDA-MEC consiga atingir a taxa de 100% de sucesso na otimização. Por outro lado, o EDA-MEC necessita de rebeldes para aumentar a taxa de sucesso na otimização de todas as demais funções. Particularmente, na mais difícil delas, a de Rosenbrock, observa-se que:

- A melhor mediana ocorre quando o número de rebeldes é igual a um; e a pior com zero rebeldes;
- Excluindo o *outlier* – representado pelo círculo próximo de zero na coluna de um rebelde – o pior dos casos com um rebelde na população é melhor do que todos os demais piores casos, isto é, o limite inferior do caso com um rebelde é superior aos outros limites inferiores das demais configurações de rebeldes;
- Com um rebelde, os resultados estão menos dispersos, concentrando-se entre o 1º e 3º quartis;

- Não utilizar rebeldes proporciona o pior resultado da mediana.

Desse modo, a utilização de rebeldes é fundamental para melhorar o desempenho do EDA-MEC.

5.4.5. Análise geral dos ensaios

Os gráficos das figuras seguintes permitem visualizar:

- i. a taxa percentual de sucesso (nas ordenadas);
- ii. o número de reinicializações do algoritmo, expressos em faixa de valores (nas abscissas, mais próximas ao eixo),

ambos resultantes de ensaios para cada uma das seis funções *benchmarks* (título do gráfico), considerando como parâmetros de entrada:

- A função cópula (subtítulo do gráfico);
- Tamanho da população (legenda de cores das barras do gráfico);
- As distribuições marginais (dividindo as abscissas em duas metades, empírica e normal);
- Número de rebeldes (expressos nas duas metades das abscissas, para as duas distribuições marginais).

Para facilitar a comparação dos resultados dos ensaios por função, os mesmos serão apresentados em sequência, sendo a análise dos mesmos feita após o último gráfico.

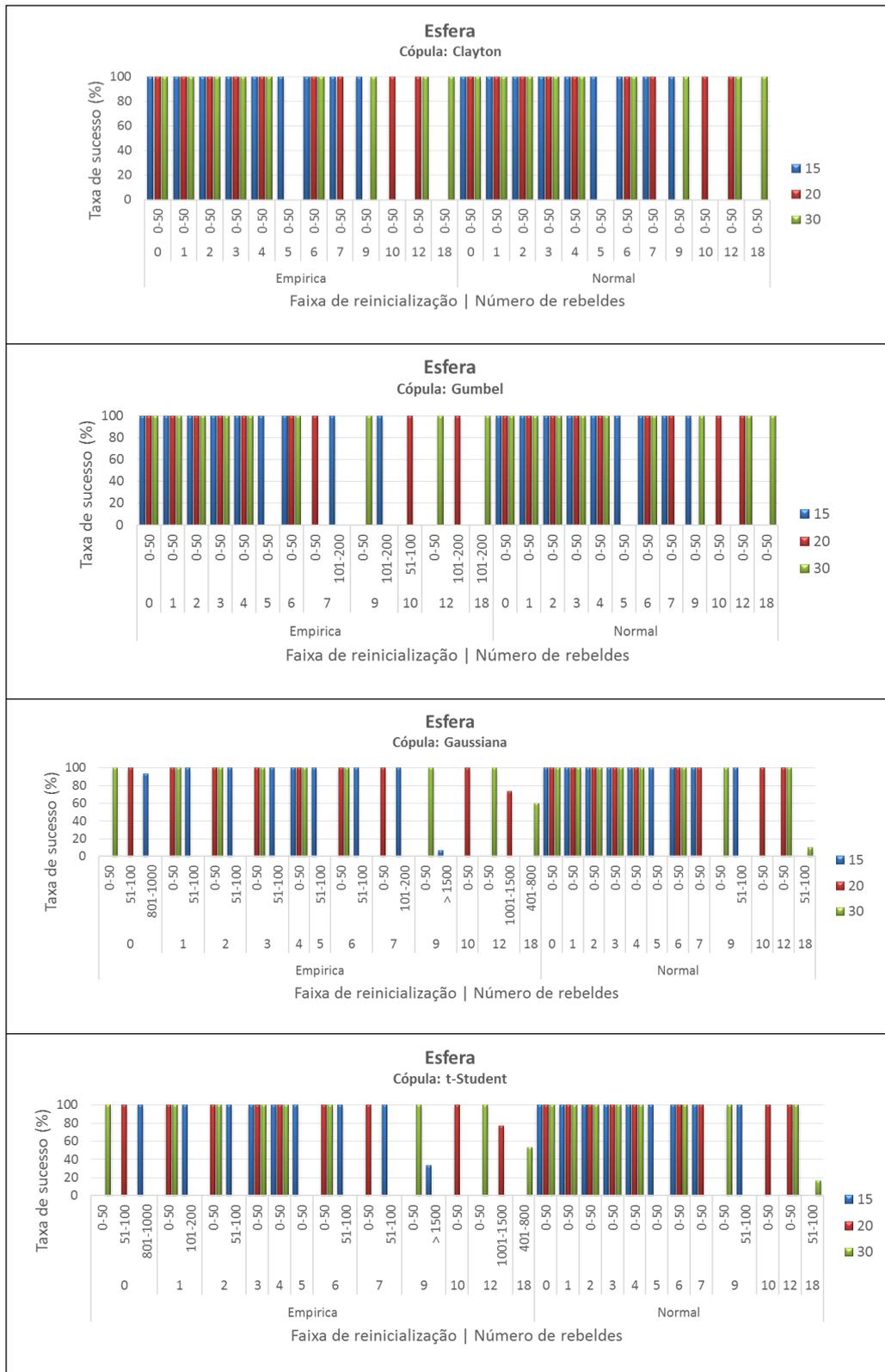


Figura 5.14 - Desempenho do EDA-MEC na otimização da função esfera, considerando todas as combinações de parâmetros.

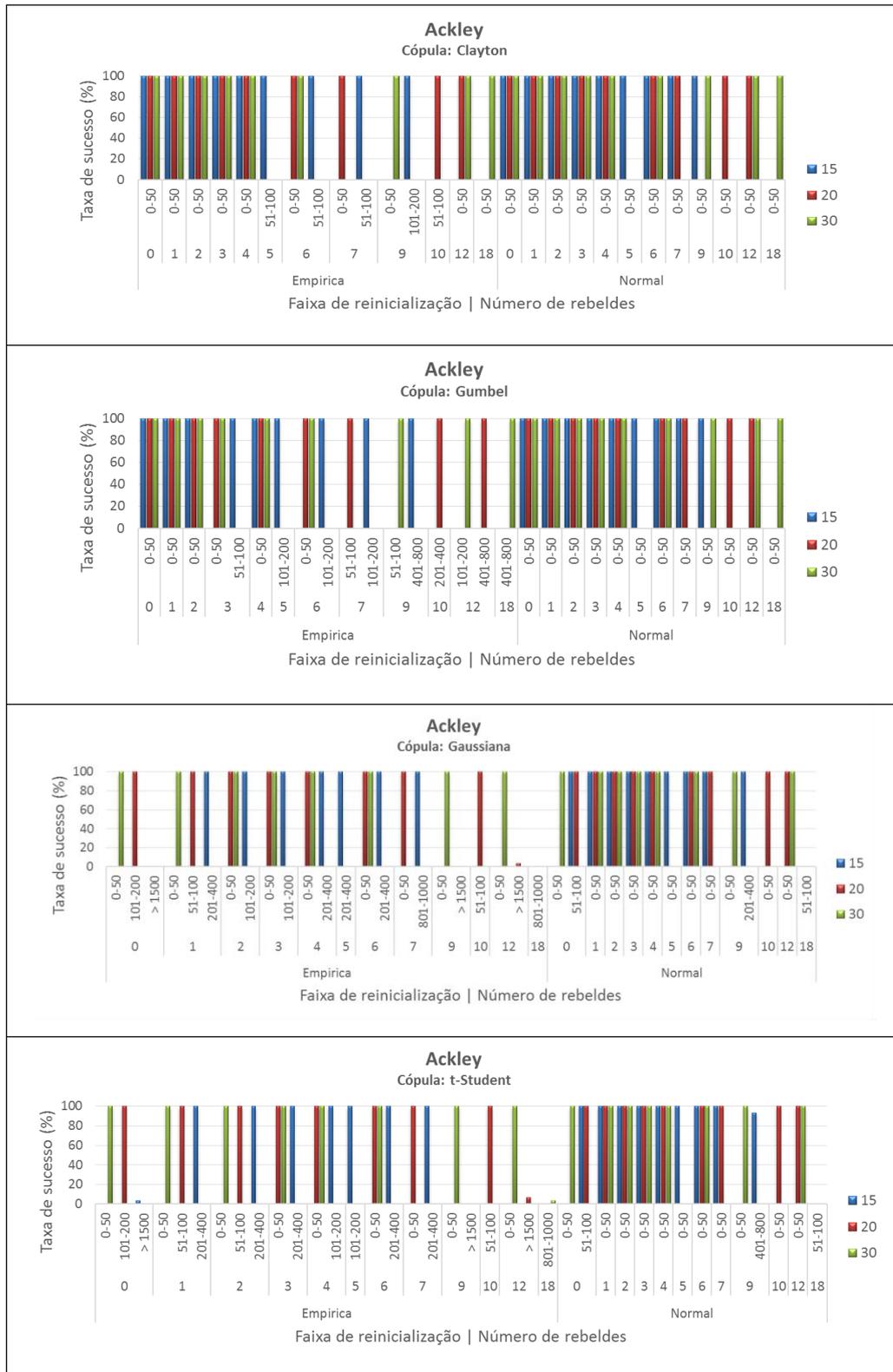


Figura 5.15 - Desempenho do EDA-MEC na otimização da função de Ackley, considerando todas as combinações de parâmetros.

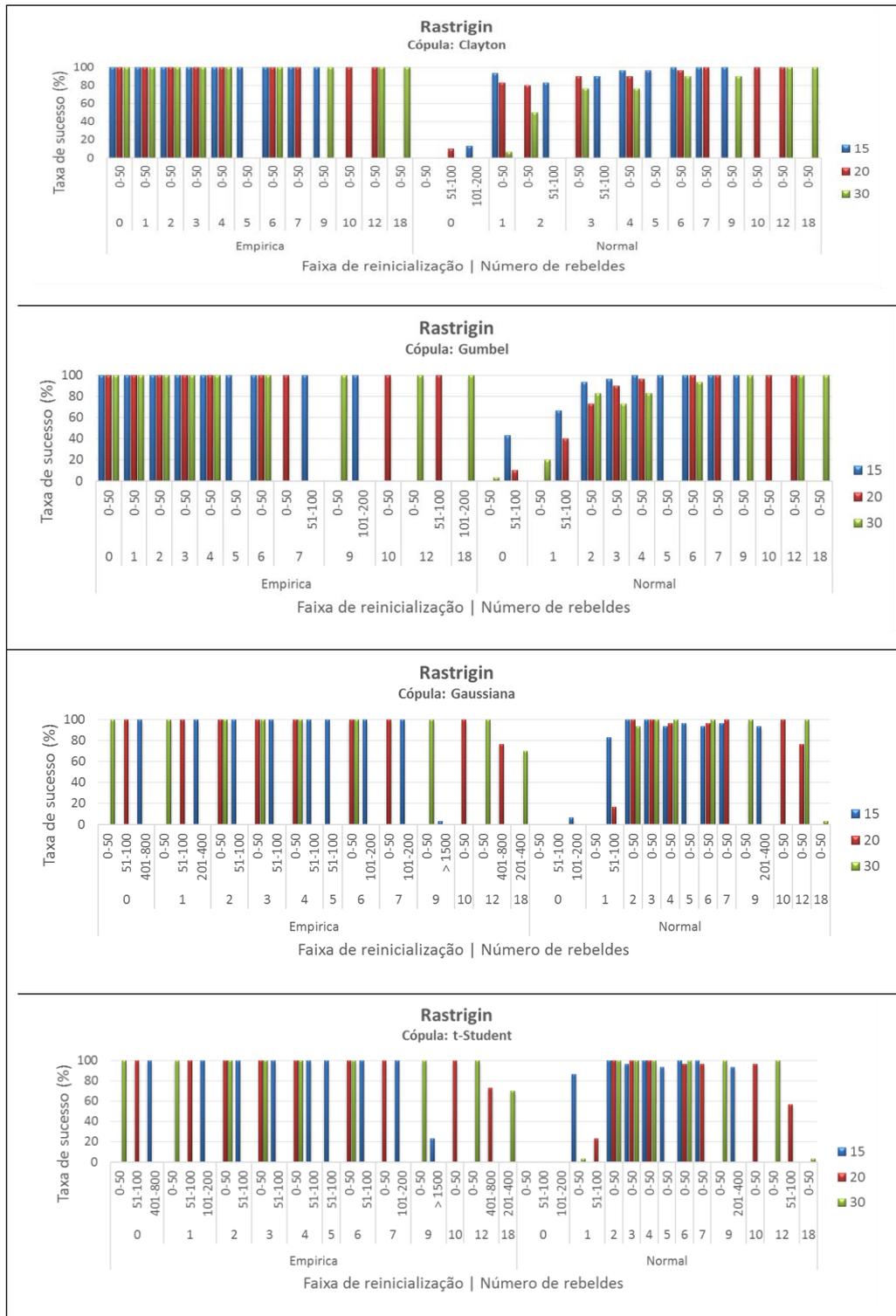


Figura 5.16 - Desempenho do EDA-MEC na otimização da função de Rastrigin, considerando todas as combinações de parâmetros.

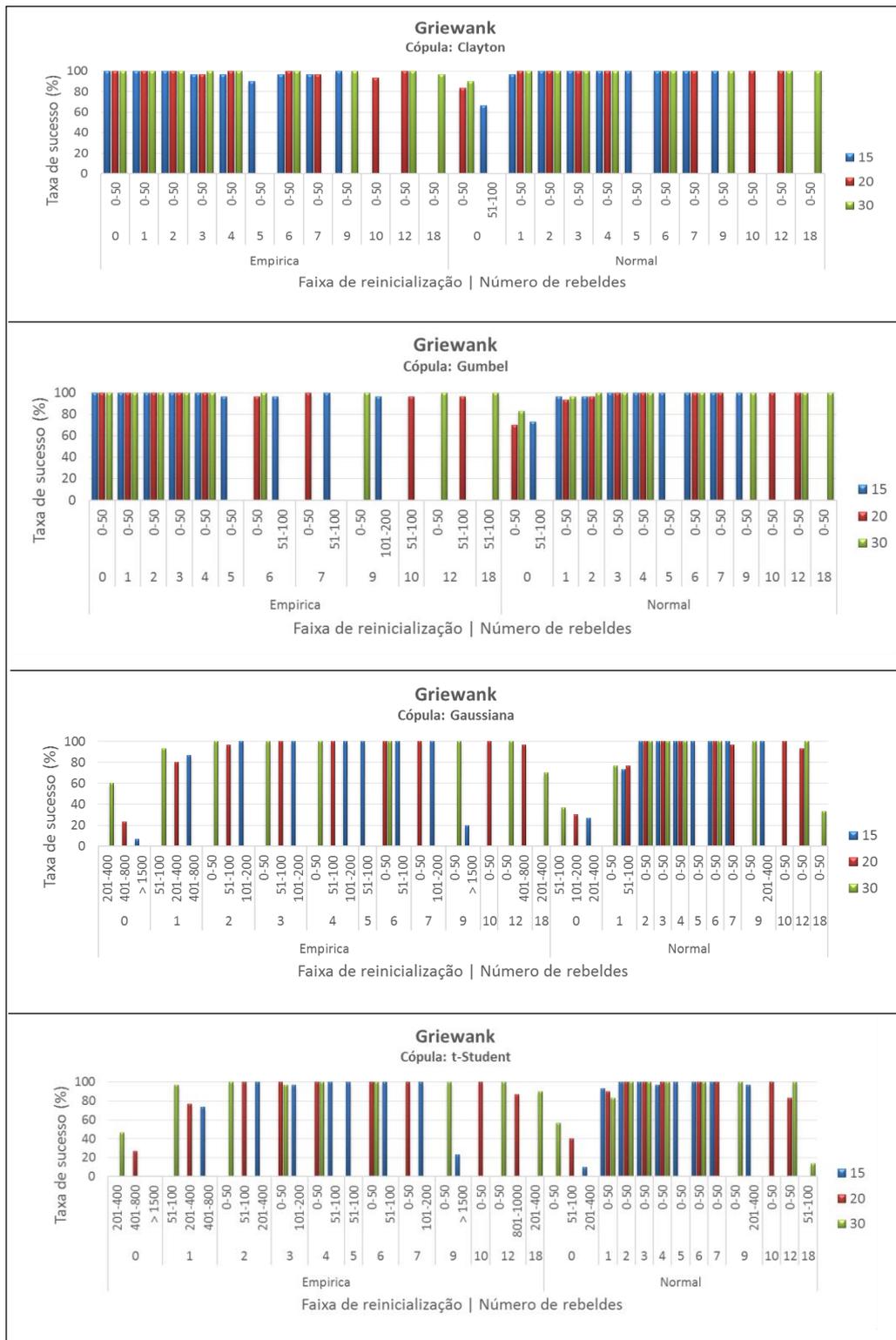


Figura 5.17 - Desempenho do EDA-MEC na otimização da função de Griewank, considerando todas as combinações de parâmetros.

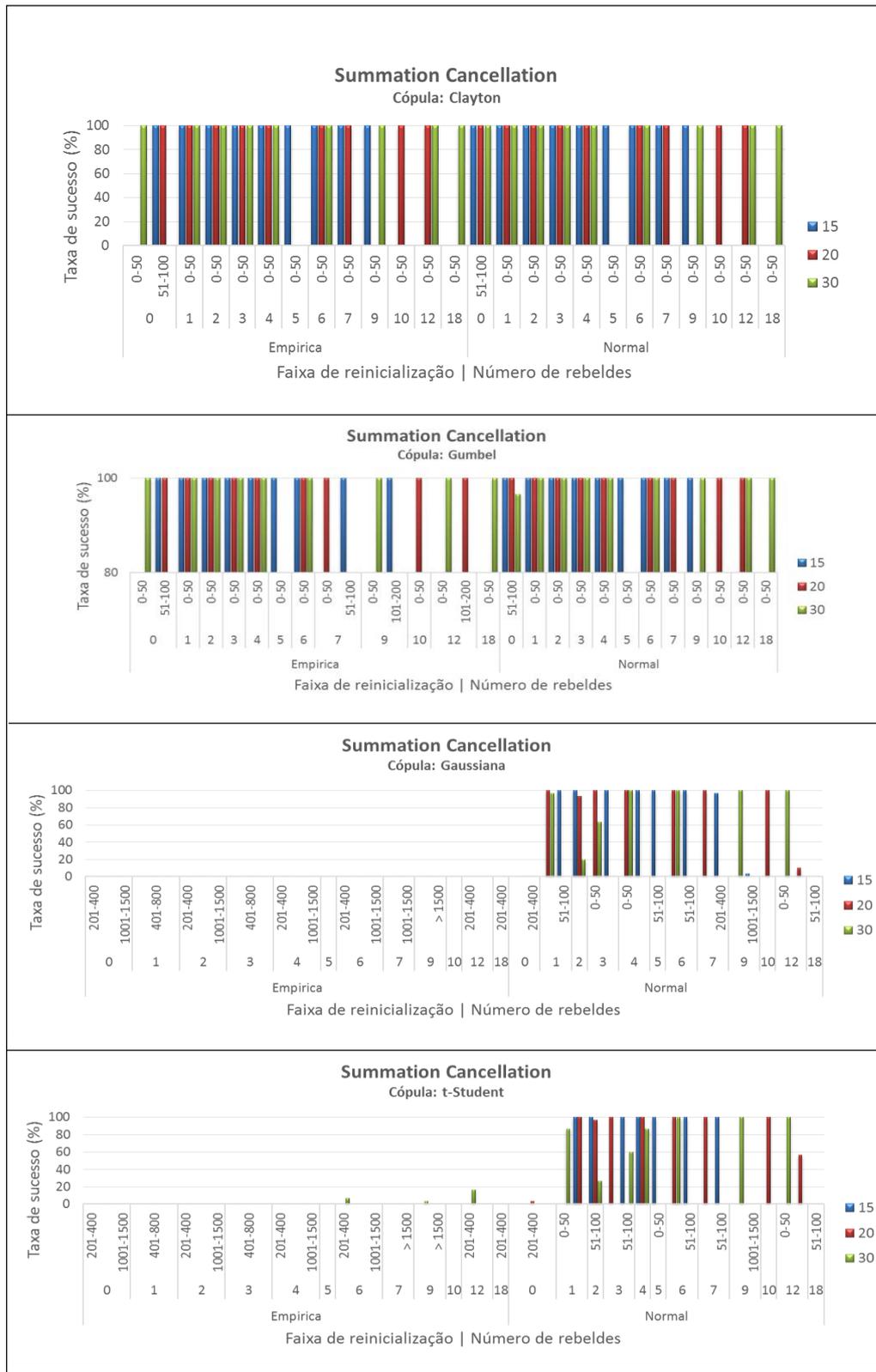


Figura 5.18 - Desempenho do EDA-MEC na otimização da função Summation Cancellation, considerando todas as combinações de parâmetros.

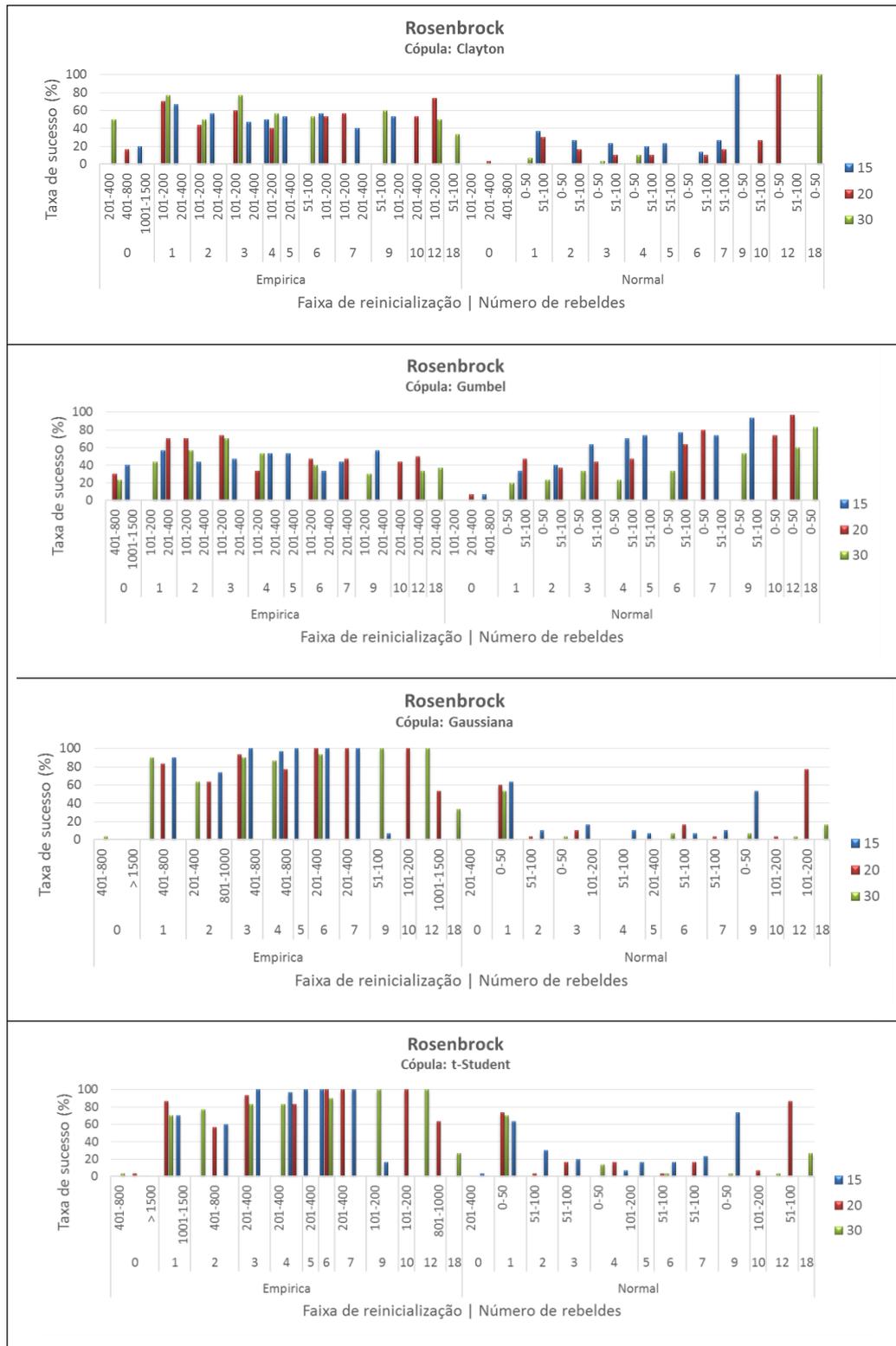


Figura 5.19 - Desempenho do EDA-MEC na otimização da função Rosenbrock, considerando todas as combinações de parâmetros.

A análise dos ensaios das funções Esfera e Ackley, nas figuras 5.14 e 5.15, pode ser feita conjuntamente dada a similaridade observada nos resultados. A modelagem através das cópulas de Clayton e Gumbel garantem a otimização das funções Esfera e Ackley com pouco esforço computacional, haja vista a baixa quantidade de reinicializações, situadas na faixa de (0-50), e de modo independente da distribuição escolhida para modelar as marginais. No caso da cópula de Gumbel e marginal empírica, verifica-se que a introdução de mais de 7 rebeldes prejudica o desempenho, visto o aumento provocado do número de reinicializações no algoritmo. Para as cópulas elípticas, identifica-se comportamento semelhante entre a Gaussiana e a t-Student: maior esforço computacional que o das arquimedianas – refletido no maior número de reinicializações – com algumas poucas falhas em ensaios que utilizam a distribuição empírica. Nas cópulas elípticas, o desempenho é melhor com a distribuição normal, pois as reinicializações ocorrem menos frequentemente, exceto na configuração de população 30 com 18 rebeldes, na qual o EDA-MEC otimiza menos de 20% dos experimentos.

De acordo com a figura 5.16, também é possível otimizar a função de Rastrigin com qualquer combinação de cópulas e distribuições marginais. Em particular, na otimização desta função é possível perceber claramente a utilidade dos rebeldes: independente da cópula escolhida, nos ensaios com marginais normais, o percentual da taxa de sucesso aumenta com a introdução de pelo menos um rebelde na população. Observa-se também que a utilização dessas marginais é sempre menos onerosa computacionalmente nas cópulas Gaussiana e t-Student, dado o menor número de reinicializações. Característica esta também observada nas cópulas arquimedianas com distribuição empírica.

Segundo a figura 5.17, de modo semelhante à função de Rastrigin, observa-se a contribuição dos rebeldes na otimização de mais uma função multimodal, a de Griewank. Percebe-se novamente, nas cópulas gaussianas, o aumento do desempenho do algoritmo, medido em termos de taxa de sucesso, quando se adiciona pelo menos um rebelde. Além disso, a introdução deles é, de fato, mais eficiente, quando a distribuição normal é considerada na modelagem das marginais, uma vez que ocorre redução do número de reinicializações na passagem de zero para um rebelde. Por outro lado, as cópulas arquimedianas com marginais empíricas demonstram não precisar de rebeldes para garantir a otimização da Griewank, haja vista atingirem o percentual máximo da taxa de sucesso com zero

rebeldes. E na sequência da análise gráfica, verifica-se que a presença de rebeldes, com a cópula de Gumbel, é sugestiva de trazer prejuízos à otimização, dado o discreto aumento do número de reinicializações observado com os ensaios a partir de 6 rebeldes.

Parte dos resultados obtidos com a Summation Cancellation, na figura 5.18, confirmam o que se verificou anteriormente na figura 5.8: a cópula de Clayton, independentemente da distribuição marginal escolhida, otimiza esta função, com baixo esforço computacional, em todos os experimentos. Ainda que amostrada pelo algoritmo de Marshall-Olkin, a cópula de Gumbel segue o mesmo desempenho da de Clayton, exceto em um ensaio, com população 30 e zero rebeldes, no qual converge para um mínimo local. Por outro lado, os resultados obtidos com as cópulas elípticas precisam ser investigados. Estas cópulas juntamente com marginais empíricas falham em todos os experimentos da Summation Cancellation, mas são bem sucedidas com marginais normais em ensaios com pelo menos um rebelde. É possível relacionar duas hipóteses para explicar o referido comportamento:

- i. a distribuição normal possui maior densidade espacial que a empírica, cobrindo todo o espaço de busca. À medida que a evolução ocorre, a distribuição empírica – baseada na população daquela geração – deixa de representar determinadas regiões do espaço de busca onde a probabilidade é zero (não há indivíduos) e por isso falha da otimização;
- ii. ao utilizar a distribuição empírica, o EDA-MEC reatribui valores de iterações anteriores (que representavam ótimos locais) para pelo menos uma variável. Nessa hipótese, não é o fato da marginal empírica estar combinada com a cópula que faz o algoritmo funcionar bem, mas sim as características da função *benchmark*. Assim, o desempenho deve ser avaliado pela combinação função *benchmark* e distribuição marginal. No entanto, não se pode descartar a influência da cópula escolhida para a otimização. Por exemplo, a escolha de uma cópula com dependência nos extremos amarra a geração de amostras o que pode não ser adequado para determinado problema.

Por fim, a mais desafiadora das funções *benchmarks* deste conjunto, a Rosenbrock, conseguiu ser otimizada mais vezes em ensaios com a combinação que gerou os piores resultados para Summation Cancellation: cópulas elípticas e

marginais empíricas. Observa-se que nesta configuração o EDA-MEC precisa fazer uso de rebeldes para alcançar o mínimo global da Rosenbrock. Embora as cópulas arquimedianas tenham obtido a melhor taxa média de sucesso, somente a cópula de Clayton conseguiu otimizar a função em todos os experimentos de três ensaios, utilizando marginais normais e populações de 15, 20 e 30 com 9, 12 e 18 rebeldes, respectivamente. Além disso, estes ensaios foram mais eficientes, uma vez que provocaram reinicializações na faixa mínima de (0-50), diferentemente das elípticas que realizaram de (51-100), no melhor dos ensaios.

5.5. Comparação com outros CEDAs e com o CMA-ES

Após os diversos ensaios realizados nas seções anteriores, que permitiram identificar a melhor parametrização do EDA-MEC por função, pode-se, então, comparar os melhores resultados obtidos com os de outros EDAs baseados em cópulas e com os do CMA-ES.

As tabelas adiante farão referência aos seguintes algoritmos implementados por GONZÁLEZ-FERNANDEZ & SOTO (2012, 2014):

- **UMDA** (*Univariate Marginal Distribution Algorithm*) para variáveis contínuas. Muito embora, não seja originalmente definido em termos de cópulas, este algoritmo é descrito como um CEDA que modela a dependência entre variáveis através do produto de cópulas multivariadas, portanto, supondo independência entre as variáveis (ver seção 2.2.3);
- **GCEDA** (*Gaussian Copula Estimation of Distribution Algorithm*). Este algoritmo é descrito como um EDA baseado em cópula normal multivariada, que permite a construção de distribuições com estrutura de dependência normal e marginais não normais.
- **CVEDA** (*Estimation of Distribution Algorithm based on Canonical Vine*). Este algoritmo é descrito como um EDA baseado em cópulas bivariadas decompostas graficamente de uma densidade de probabilidade multivariada, dos modelos *pair-copulas* (ver seção 3.4.2).

- DVEDA (*Estimation of Distribution Algorithm based on Drawable Vine*). Este algoritmo é descrito como um EDA baseado em cópulas bivariadas decompostas graficamente de uma densidade de probabilidade multivariada, dos modelos *pair-copulas* (ver seção 3.4.2).
- Copula MIMIC (*Copula Mutual Information Maximization for Input Clustering*). Este algoritmo é uma extensão do MIMIC (EDA de dependência aos pares) para o domínio contínuo. Este modelo adaptado utiliza cópulas bivariadas, ao invés de distribuições normais bivariadas, para modelar dependências (ver seção 3.4.2).

No estudo empírico realizado por GONZÁLEZ-FERNANDEZ & SOTO (2012) apenas relações de independência e lineares são consideradas. Então, somente cópulas produto e normal foram utilizadas nestes EDAs.

Relativamente aos resultados do CMA-ES, foi utilizado o código fonte (versão 3.61.beta) disponibilizado no website do autor do método⁶. Para estabelecer condições justas de comparação com o EDA-MEC e com os algoritmos de GONZÁLEZ-FERNANDEZ & SOTO (2012), configurações semelhantes foram adotadas, sendo desabilitados diversos critérios de parada que o código fornece. Mesmo assim, restrições ocultas do mesmo interromperam a execução antes das 300.000 avaliações, nas situações em que o CMA-ES estagnava. Nas tabelas seguintes, isto significa que, nos ensaios que o método falhou, isto é, quando a taxa de sucesso não foi 30/30, o número de avaliações (AES) certamente é maior do que o informado na tabela.

A modelagem do EDA-MEC tomou por base a cópula de Clayton, variando as marginais, tamanho de população e rebeldes de modo a obter o melhor resultado para cada função *benchmark*. Abaixo de cada tabela, são informadas as marginais e número de rebeldes utilizados e o número de reinicializações ocorrido.

Nas tabelas comparativas a seguir, estão destacados em negrito o modelo apresentado nesta tese (**EDA-MEC**) e em sombreado o que obteve o melhor desempenho naquela função. O critério do melhor desempenho considera nesta ordem: i) melhor MBF (*Mean Best Fitness* – melhor aptidão média); ii) menor AES (*Average Number of Evaluations to a Solution* – número médio de avaliações).

⁶ https://www.lri.fr/~hansen/cmaes_inmatlab.html#matlab

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|-----------------|-----------|-----------------|----------------|---------------|------------------------|----------------------|--------------------|
| EDA-MEC* | 15 | 30/30 | 1,3E-8 | 9,9E-7 | 5,4E-7 ± 3,1E-7 | 1.647 ± 1.670 | 1,23 ± 1,06 |
| UMDA | 81 | 30/30 | – | – | 6,9E-7 ± 2,3E-7 | 3.823 ± 128,3 | 0,4 ± 0,0 |
| GCEDA | 310 | 30/30 | – | – | 6,5E-7 ± 2,0E-7 | 13.082 ± 221,4 | 0,9 ± 0,0 |
| CVEDA | 104 | 30/30 | – | – | 6,7E-7 ± 1,8E-7 | 4.777 ± 118,8 | 8,3 ± 1,5 |
| DVEDA | 104 | 30/30 | – | – | 6,7E-7 ± 2,0E-7 | 4.787 ± 100,2 | 8,2 ± 1,1 |
| Cópula MIMIC | 150 | 30/30 | – | – | 6,9E-7 ± 1,7E-7 | 6.495 ± 209,0 | 468,8 ± 62,9 |
| CMA-ES | 10 | 30/30 | 2,9E-7 | 9,9E-7 | 7,7E-7 ± 1,7E-7 | 2.119 ± 96 | 0,29 ± 0,02 |

* Marginal: empírica; número de rebeldes = 7; número médio de reinicializações: 14 ± 37

Tabela 5.1 - Função Esfera

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|-----------------|-----------|-----------------|----------------|---------------|------------------------|----------------------|--------------------|
| EDA-MEC* | 20 | 30/30 | 4,8E-7 | 9,9E-7 | 8,2E-7 + 1,4E-7 | 3.016 + 1.874 | 1,93 + 1,09 |
| UMDA | 81 | 30/30 | – | – | 8,1E-7 ± 1,1E-7 | 4.998 ± 88,0 | 0,6 ± 0,0 |
| GCEDA | 279 | 30/30 | – | – | 8,3E-7 ± 1,1E-7 | 15.633 ± 258,8 | 1,6 ± 0,0 |
| CVEDA | 104 | 30/30 | – | – | 8,1E-7 ± 1,0E-7 | 6.330 ± 163,2 | 10,6 ± 1,8 |
| DVEDA | 111 | 30/30 | – | – | 7,9E-7 ± 1,4E-7 | 6.679 ± 133,8 | 11,4 ± 1,6 |
| Cópula MIMIC | 188 | 30/30 | – | – | 8,0E-7 ± 1,2E-7 | 10.784 ± 143,7 | 800,1 ± 122,1 |
| CMA-ES | 10 | 29/30 | 8,7E-7 | 1,9E+1 | 1,9E+1 ± 3,5E+0 | 25.245 ± 2.595 | 1,02 ± 0,02 |

* Marginal: empírica; número de rebeldes = 7; número médio de reinicializações: 10 ± 22

Tabela 5.2 - Função de Ackley

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|----------------|-----------|-----------------|----------------|---------------|------------------------|----------------------|--------------------|
| EDA-MEC | 15 | 30/30 | 3,2E-7 | 9,8E-7 | 7,4E-7 ± 1,8E-7 | 3.217 ± 2.869 | 2,85 ± 2,57 |
| UMDA | 447 | 30/30 | – | – | 6,7E-7 ± 2,3E-7 | 33.614 ± 2.452 | 1,7 ± 0,1 |
| GCEDA | 721 | 30/30 | – | – | 6,8E-7 ± 1,8E-7 | 46.095 ± 2.158 | 2,8 ± 0,1 |
| CVEDA | 447 | 30/30 | – | – | 6,6E-7 ± 1,7E-7 | 32.914 ± 2.011 | 44,8 ± 13,3 |
| DVEDA | 325 | 30/30 | – | – | 7,3E-7 ± 1,7E-7 | 24.710 ± 1.754 | 31,3 ± 6,9 |
| Cópula MIMIC | 386 | 30/30 | – | – | 6,4E-7 ± 2,1E-7 | 27.315 ± 1.673,9 | 1.432 ± 218,6 |
| CMA-ES | 10 | 27/30 | 5,6E-7 | 1,3E+2 | 4,4E+1 ± 3,9E+1 | 226.559 ± 93.676 | 7,66 ± 7,3 |

* Marginal: empírica; número de rebeldes = 4; número médio de reinicializações: 9 ± 12

Tabela 5.3 - Função de Rastrigin

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|----------------|-----------|-----------------|----------------|---------------|-------------------------|----------------------|--------------------|
| EDA-MEC | 15 | 30/30 | 6,9E-8 | 9,8E-7 | 6,34E-7 ± 2,6E-7 | 2.665 ± 2.247 | 2,01 ± 1,66 |
| UMDA | 111 | 30/30 | – | – | 6,6E-7 ± 1,9E-7 | 5.224 ± 231,2 | 0,5 ± 0,0 |
| GCEDA | 355 | 30/30 | – | – | 6,9E-7 ± 1,8E-7 | 15.099 ± 414,1 | 1,2 ± 0,0 |
| CVEDA | 142 | 30/30 | – | – | 7,0E-7 ± 1,8E-7 | 6.579 ± 389,9 | 9,3 ± 2,0 |
| DVEDA | 150 | 30/30 | – | – | 6,5E-7 ± 2,4E-7 | 6.785 ± 338,1 | 9,5 ± 1,9 |
| Cópula MIMIC | 188 | 30/30 | – | – | 6,6E-7 ± 1,8E-7 | 8.221 ± 220,4 | 595,7 ± 91,6 |
| CMA-ES | 10 | 30/30 | 2,9E-7 | 9,9E-7 | 7,6E-7 ± 1,7E-7 | 76.099 ± 83.741 | 0,49 ± 0,19 |

* Marginal: empírica; número de rebeldes = 9; número médio de reinicializações: 25 ± 35

Tabela 5.4 - Função de Griewank

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|----------------|-----------|-----------------|----------------|--------------|-----------------------|----------------------|-------------------|
| EDA-MEC | 15 | 30/30 | -1E+5 | -1E+5 | -1E+5 ± 1,5E-7 | 6.104 ± 1.488 | 5,2 ± 1,27 |
| UMDA | 2000 | 0/30 | – | – | -5,7E+2 ± 3,4E+2 | 300.000 ± 0,0 | 66,8 ± 0,7 |
| GCEDA | 355 | 30/30 | – | – | -1E+5 ± 1,3E-7 | 42.434 ± 305 | 9,3 ± 0,4 |
| CVEDA | 325 | 30/30 | – | – | -1E+5 ± 1,3E-7 | 44.622 ± 858 | 537,2 ± 6,6 |
| DVEDA | 965 | 30/30 | – | – | -1E+5 ± 9,3E-8 | 117.408 ± 959 | 2.367 ± 20 |
| Cópula MIMIC | 2000 | 30/30 | – | – | -2,3E+4 ± 2,7E+4 | 300.000 ± 0,0 | 10.426 ± 1054 |
| CMA-ES | 10 | 30/30 | -1E+5 | -1E+5 | -1E+5 ± 7,0E-8 | 7.483 ± 5.004 | 1,0 ± 0,67 |

* Marginal: empírica; número de rebeldes = 9; número médio de reinicializações: 13 ± 7

Tabela 5.5 - Função Summation Cancellation

| Algoritmo | Tam. Pop | Taxa de Sucesso | Melhor Aptidão | Pior Aptidão | MBF | AES | Tempo CPU (s) |
|----------------|-----------|-----------------|----------------|---------------|------------------------|------------------------|-------------------|
| EDA-MEC | 20 | 30/30 | 9,9E-8 | 9,9E-7 | 5,5E-7 ± 2,5E-7 | 14.367 ± 13.001 | 9,5 ± 8,64 |
| UMDA | 2000 | 0/30 | – | – | 8,0E+0 ± 2,6E-2 | 300.000 ± 0,0 | 72,0 ± 0,1 |
| GCEDA | 2000 | 0/30 | – | – | 7,5E+0 ± 1,9E-1 | 300.000 ± 0,0 | 74,3 ± 0,4 |
| CVEDA | 2000 | 0/30 | – | – | 7,5E+0 ± 1,1E-1 | 193.867 ± 48.243 | 1.279 ± 532 |
| DVEDA | 2000 | 0/30 | – | – | 7,5E+0 ± 1,5E-1 | 172.200 ± 35.184 | 961 ± 386 |
| Cópula MIMIC | 2000 | 0/30 | – | – | 7,6E+0 ± 1,3E-1 | 139.000 ± 5.139 | 6.174 ± 821 |
| CMA-ES | 10 | 30/30 | 8,2E-7 | 1,0E-6 | 8,2E-7 ± 1,6E-7 | 6.212 ± 1.103 | 0,89 ± 0,16 |

* Marginal: normal; número de rebeldes = 12; número médio de reinicializações: 12 ± 13

Tabela 5.6 - Função de Rosenbrock

Como resultado geral, o EDA-MEC otimizou todas as funções, sendo que somente não obteve o melhor desempenho na função de Rosenbrock, para a qual o CMA-ES o fez com menos da metade do número de avaliações do EDA-MEC.

Na Esfera, Ackley e Griewank, nas quais não existem dependências fortes entre as variáveis dessas funções, todos os métodos alcançaram o mínimo global em todos os experimentos, exceto o CMA-ES, na função de Ackley, falhando em um experimento. É importante ressaltar que o código do CMA-ES exige somente duas parametrizações do usuário: a função a ser otimizada e seu número de variáveis. Até mesmo o tamanho da população é função do número de variáveis. No entanto, um parâmetro opcional de reinicialização, que dobra o tamanho da população original na continuação do experimento, foi habilitado nos experimentos de todas as funções. Verificou-se que sem ele a Rosenbrock falhava em algum experimento, mas convergia em todos com três. Mesmo assim, essa configuração não surtiu efeito para a de Ackley e nem para a de Rastrigin, na qual o CMA-ES falhou em três experimentos.

Ainda em relação às funções Esfera, de Ackley e de Griewank, entre os algoritmos de GONZÁLEZ-FERNANDEZ & SOTO (2012), o UMDA teve o melhor desempenho; os outros exigiram o cálculo de um número maior de parâmetros para representar as relações entre as variáveis. Portanto, populações maiores foram necessárias para calculá-las de forma confiável, segundo os autores.

Na função de Rastrigin, que não é um problema em que as interações entre as variáveis desempenham um papel importante para o sucesso na otimização, realizando um número de avaliações diferente, todos os modelos alcançaram o ótimo global, exceto o CMA-ES. Nesta função, GONZÁLEZ-FERNANDEZ & SOTO (2012) observaram que a combinação de cópulas produto e normal em um único modelo probabilístico é melhor que assumir independência ou uma estrutura de dependência linear multivariada, justificando o desempenho melhor do DVEDA e do Copula MIMIC sobre os demais algoritmos dos autores.

Na Summation Cancellation, uma correta representação das interações fortes entre as variáveis é essencial para um algoritmo otimizá-la. Assim, falham os algoritmos UMDA, que ignora essa necessidade ao assumir independência entre as variáveis, e o Copula MIMIC, cujo modelo não consegue representar as dependências para sucesso na otimização.

E finalmente, na Rosenbrock, um problema reconhecidamente difícil para algoritmos de otimização, com dependência não linear entre as variáveis, apenas o EDA-MEC e o CMA-ES conseguiram otimizá-la.

5.6. Escalabilidade do EDA-MEC

A fim de verificar o desempenho do algoritmo com o aumento do número de variáveis, o EDA-MEC simulou a otimização das seis funções *benchmarks* com 50 e 100 variáveis, utilizando os mesmos parâmetros da subseção anterior. A tabela seguinte apresenta o resultado desses ensaios, em termos de taxa de sucesso, média de reinicializações, melhor aptidão média, número médio de avaliações e média de tempo de máquina em 30 experimentos independentes. Para facilitar a comparação, os resultados dos ensaios com 10 variáveis são reproduzidos na mesma tabela.

| Função | Num Variáveis | Tam. Pop | Num Rebel. | Taxa de Sucesso | Média de Reinic. | MBF | AES | Tempo CPU (s) |
|------------------------|---------------|----------|------------|-----------------|------------------|-----------------|-----------------|-----------------|
| Esfera | 10 | 15 | 7 | 30/30 | 14 ± 37 | 5,4E-7 ± 3,1E-7 | 1.647 ± 1.670 | 1,23 ± 1,06 |
| | 50 | 15 | 7 | 30/30 | 8 ± 9 | 7,5E-7 ± 2,2E-7 | 1.560 ± 710 | 15,51 ± 6,71 |
| | 100 | 15 | 7 | 30/30 | 12 ± 14 | 6,9E-7 ± 2,5E-7 | 2.085 ± 1.005 | 70,91 ± 32,41 |
| Ackley | 10 | 20 | 7 | 30/30 | 10 ± 22 | 8,2E-7 ± 1,4E-7 | 3.016 ± 1.874 | 1,93 ± 1,09 |
| | 50 | 20 | 7 | 30/30 | 7 ± 6 | 9,0E-7 ± 1,1E-7 | 3.480 ± 1.014 | 31,63 ± 9,05 |
| | 100 | 20 | 7 | 30/30 | 13 ± 26 | 9,0E-7 ± 1,0E-7 | 5.248 ± 3.388 | 168,52 ± 103,13 |
| Rastrigin | 10 | 15 | 4 | 30/30 | 9 ± 12 | 7,4E-7 ± 1,8E-7 | 3.217 ± 2.869 | 2,85 ± 2,57 |
| | 50 | 15 | 4 | 30/30 | 41 ± 43 | 8,0E-7 ± 2,1E-7 | 11.810 ± 14.815 | 117,46 ± 147,98 |
| | 100 | 15 | 4 | 30/30 | 58 ± 58 | 8,4E-7 ± 1,1E-7 | 14.529 ± 12.840 | 550,41 ± 487,17 |
| Griewank | 10 | 15 | 9 | 30/30 | 14 ± 37 | 6,3E-7 ± 2,6E-7 | 2.665 ± 2.247 | 2,01 ± 1,66 |
| | 50 | 15 | 9 | 30/30 | 8 ± 16 | 6,7E-7 ± 2,6E-7 | 1.293 ± 1.063 | 13,94 ± 10,81 |
| | 100 | 15 | 9 | 30/30 | 10 ± 35 | 6,1E-7 ± 2,4E-7 | 1.393 ± 2.057 | 53,55 ± 72,95 |
| Summation Cancellation | 10 | 15 | 9 | 30/30 | 13 ± 7 | -1E+5 ± 1,5E-7 | 6.104 ± 1.488 | 5,2 ± 1,27 |
| | 50 | 15 | 9 | 30/30 | 20 ± 4 | -1E+5 ± 2,2E-7 | 15.037 ± 3.744 | 149,54 ± 37,67 |
| | 100 | 15 | 9 | 30/30 | 24 ± 5 | -1E+5 ± 1,49E-7 | 21.574 ± 6.742 | 761,56 ± 236,62 |
| Rosenbrock* | 10 | 20 | 12 | 30/30 | 12 ± 13 | 5,5E-7 ± 2,5E-7 | 14.367 ± 13.001 | 9,5 ± 8,64 |
| | 50 | 20 | 12 | 30/30 | 15 ± 11 | 6,4E-7 ± 2,5E-7 | 20.385 ± 15.150 | 171,33 ± 127,96 |
| | 100 | 20 | 12 | 30/30 | 14 ± 11 | 5,5E-7 ± 2,2E-7 | 19.670 ± 15.164 | 620,61 ± 478,34 |

* Marginal: normal; todas as demais utilizaram distribuições marginais empíricas.

Tabela 5.7 - Ensaios de escalabilidade do EDA-MEC

Observa-se que o EDA-MEC alcança o mínimo global das seis funções *benchmarks* em todos os experimentos, sem realizar o número máximo de 300.000 avaliações, mas com um crescimento exponencial do tempo de CPU. Este resultado demonstra a estabilidade numérica do algoritmo, sugerindo que o mesmo pode otimizar funções com mais de 100 variáveis de decisão.

5.7. Outros ensaios

Os ensaios dessa seção buscam verificar o desempenho do EDA-MEC quando os graus de liberdade (da cópula t-Student) e o parâmetro das cópulas arquimedianas são obtidos por MLE.

Na tabela 5.8, são re-exibidos dados de ensaios das seções anteriores que resultaram no melhor desempenho, em termos de número médio de avaliações (AES) e tempo de CPU, com as cópulas Gaussiana e t-Student para as seis funções *benchmarks* com 10 variáveis. Nas duas primeiras linhas de cada função, são observados desempenhos muito semelhantes entre as duas cópulas, ainda que os graus de liberdade fixos e de baixo valor tenham sido propositalmente parametrizados para que a distribuição t não convergisse para a Gaussiana. Isto pode ser explicado pela característica em comum das funções *benchmarks*: ótimo global localizado no centro do domínio. Note que a diferença maior de desempenho ocorreu na Rosenbrock, que tem não possui ótimo global central em domínio assimétrico. No que diz respeito ao emprego do MLE no cálculo dos graus de liberdade a cada geração do EDA-MEC, nota-se redução no AES, exceto na Esfera e Rosenbrock, que não compensa o aumento considerável de tempo computacional

| Função | Cópula | G.L. | Tam. Pop | Num Rebel. | Taxa de Sucesso | Média de Reinic. | MBF | AES | Tempo CPU (s) |
|------------------------|-----------|------|----------|------------|-----------------|------------------|-----------------|-----------------|-----------------|
| Esfera | Gaussiana | – | 20 | 10 | 30/30 | 0,1 ± 0,6 | 5,7E-7 ± 2,0E-7 | 568 ± 354 | 0,10 ± 0,06 |
| | t-Student | 10 | 20 | 10 | 30/30 | 0,1 ± 0,3 | 6,7E-7 ± 2,2E-7 | 506 ± 105 | 0,11 ± 0,03 |
| | t-Student | MLE | 20 | 10 | 30/30 | 0,3 ± 0,7 | 5,9E-7 ± 2,8E-7 | 525 ± 308 | 12,03 ± 7,06 |
| Ackley | Gaussiana | – | 30 | 12 | 30/30 | 0 ± 0 | 8,0E-7 ± 1,5E-7 | 1.051 ± 50 | 0,13 ± 0,01 |
| | t-Student | 15 | 30 | 12 | 30/30 | 0 ± 0 | 8,0E-7 ± 1,5E-7 | 1.004 ± 41 | 0,13 ± 0,01 |
| | t-Student | MLE | 30 | 12 | 30/30 | 0 ± 0 | 7,7E-7 ± 1,5E-7 | 877 ± 30 | 22,9 ± 1,34 |
| Rastrigin* | Gaussiana | – | 20 | 7 | 30/30 | 32 ± 17 | 7,0E-7 ± 2,5E-7 | 7.466 ± 3.499 | 2,53 ± 1,19 |
| | t-Student | 10 | 20 | 7 | 30/30 | 29 ± 15 | 6,6E-7 ± 2,7E-7 | 7.076 ± 3.125 | 2,09 ± 0,93 |
| | t-Student | MLE | 20 | 7 | 30/30 | 23 ± 11 | 7,8E-7 ± 2,3E-7 | 5.649 ± 1.962 | 169,94 ± 58,19 |
| Griewank | Gaussiana | – | 15 | 7 | 30/30 | 3 ± 3 | 6,9E-7 ± 2,3E-7 | 3.431 ± 3.547 | 0,90 ± 0,94 |
| | t-Student | 7 | 15 | 7 | 30/30 | 3 ± 5 | 6,8E-7 ± 2,3E-7 | 3.211 ± 5.770 | 0,91 ± 1,66 |
| | t-Student | MLE | 15 | 7 | 30/30 | 3 ± 4 | 6,5E-7 ± 2,4E-7 | 1.825 ± 2.959 | 50,49 ± 83,61 |
| Summation Cancellation | Gaussiana | – | 20 | 10 | 30/30 | 8 ± 6 | -1E+5 ± 1,5E-7 | 14.336 ± 9.426 | 2,68 ± 1,79 |
| | t-Student | 10 | 20 | 10 | 30/30 | 7 ± 5 | -1E+5 ± 2,1E-7 | 14.442 ± 8.698 | 2,82 ± 1,72 |
| | t-Student | MLE | 20 | 10 | 30/30 | 16 ± 9 | -1E+5 ± 1,8E-7 | 20.072 ± 9.451 | 460,77 ± 213,45 |
| Rosenbrock* | Gaussiana | – | 15 | 5 | 30/30 | 360 ± 195 | 5,8E-7 ± 2,7E-7 | 40.219 ± 22.096 | 15,04 ± 8,42 |
| | t-Student | 7 | 15 | 5 | 30/30 | 343 ± 198 | 6,5E-7 ± 2,5E-7 | 38.030 ± 23.151 | 15,02 ± 9,35 |
| | t-Student | MLE | 15 | 5 | 30/30 | 332 ± 281 | 6,3E-7 ± 2,9E-7 | 35.549 ± 30.493 | 827,60 ± 721,69 |

* Marginal: empírica; todas as demais utilizaram distribuições marginais normais.

Tabela 5.8 - Desempenho do EDA-MEC com a cópula Gaussiana e a t-Student com e sem o emprego do MLE no cálculo dos graus de liberdade (G.L.) em funções *benchmarks* com 10 variáveis.

ocorrido. Deste modo, valida-se a metodologia adotada nos ensaios dessa tese, a de utilizar um número fixo e pequeno de graus de liberdade.

Na tabela seguinte estão dispostos os melhores resultados da otimização das funções *benchmarks* com 10 variáveis com a cópula de Clayton. Na primeira linha de cada função estão reproduzidos os dados da seção 5.5 e na segunda, resultados de novos ensaios, considerando o emprego do MLE no cálculo do parâmetro θ a cada geração do EDA-MEC. Neste caso, novamente verifica-se redução no AES, exceto na de Rastrigin, que não compensa o correspondente tempo de processamento. Este resultado valida a metodologia adotada nos ensaios dessa tese, a de utilizar uma estimativa do parâmetro θ pelo método da inversão do tau de Kendall.

| Função | Estimação do θ (Clayton) | Tam. Pop | Num Rebel. | Taxa de Sucesso | Média de Reinic. | MBF | AES | Tempo CPU (s) |
|------------------------|---------------------------------|----------|------------|-----------------|------------------|---------------------|---------------------|---------------------|
| Esfera | Inv.tau Kendall | 15 | 7 | 30/30 | 14 \pm 37 | 5,4E-7 \pm 3,1E-7 | 1.647 \pm 1.670 | 1,23 \pm 1,06 |
| | MLE | 15 | 7 | 30/30 | 11 \pm 17 | 7,1E-7 \pm 2,4E-7 | 1.360 \pm 901 | 11,20 \pm 17,04 |
| Ackley | Inv.tau Kendall | 20 | 7 | 30/30 | 10 \pm 22 | 8,2E-7 \pm 1,4E-7 | 3.016 \pm 1.874 | 1,93 \pm 1,09 |
| | MLE | 20 | 7 | 30/30 | 6 \pm 5 | 8,1E-7 \pm 1,9E-7 | 2.885 \pm 1.279 | 124,75 \pm 59,00 |
| Rastrigin | Inv.tau Kendall | 15 | 4 | 30/30 | 9 \pm 12 | 7,4E-7 \pm 1,8E-7 | 3.217 \pm 2.869 | 2,85 \pm 2,57 |
| | MLE | 15 | 4 | 30/30 | 18 \pm 15 | 6,5E-7 \pm 2,4E-7 | 3.844 \pm 2.889 | 159,69 \pm 129,43 |
| Griewank | Inv.tau Kendall | 15 | 9 | 30/30 | 14 \pm 37 | 6,3E-7 \pm 2,6E-7 | 2.665 \pm 2.247 | 2,01 \pm 1,66 |
| | MLE | 15 | 9 | 30/30 | 13 \pm 17 | 6,3E-7 \pm 2,6E-7 | 1.647 \pm 1.386 | 46,36 \pm 38,25 |
| Summation Cancellation | Inv.tau Kendall | 15 | 9 | 30/30 | 13 \pm 7 | -1E+5 \pm 1,5E-7 | 6.104 \pm 1.488 | 5,2 \pm 1,27 |
| | MLE | 15 | 9 | 30/30 | 12 \pm 4 | -1E+5 \pm 1,9E-7 | 5.903 \pm 1.411 | 187,22 \pm 43,44 |
| Rosenbrock* | Inv.tau Kendall | 20 | 12 | 30/30 | 12 \pm 13 | 5,5E-7 \pm 2,5E-7 | 14.367 \pm 13.001 | 9,5 \pm 8,64 |
| | MLE | 20 | 12 | 30/30 | 10 \pm 8 | 4,4E-7 \pm 2,1E-7 | 13.000 \pm 12.770 | 298,95 \pm 298,24 |

* Marginal: normal; todas as demais utilizaram distribuições marginais empíricas.

Tabela 5.9 - Desempenho do EDA-MEC com a cópula de Clayton com parâmetro obtido pelo método pelo método da inversão do tau de Kendall e por MLE em funções *benchmarks* de 10 variáveis.

6

Conclusões e trabalhos futuros

Este trabalho apresentou uma proposta de um novo modelo de otimização meta-heurística capaz de identificar boas soluções, sem grandes esforços computacionais, através da utilização eficiente de informações sobre o espaço de buscas. Para isto, foi implementado um algoritmo de estimação de distribuição (EDA – *Estimation of Distribution Algorithm*) baseado em expansão multivariada de cópulas, o EDA-MEC (*Estimation of Distribution Algorithm based on Multivariate Extension of Copulas*).

A abordagem do EDA-MEC utilizou técnicas que não são frequentemente adotadas em outros EDAs baseados em cópulas (CEDAs). Uma delas é a atualização periódica dos parâmetros da cópula. A outra é relacionada à modelagem das distribuições marginais.

Na literatura, a maioria dos CEDAs não realiza atualização periódica dos parâmetros das cópulas porque utilizam o método da máxima verossimilhança (MLE – *Maximum Likelihood Estimation*) para estimá-los. Fazendo uso de técnicas menos precisas, mas menos onerosas computacionalmente, que o MLE, o EDA-MEC estima, a cada geração, os parâmetros das cópulas arquimedianas pelo método da inversão do tau de Kendall, e a matriz de correlação, para as cópulas elípticas, aumentando a confiabilidade da informação de dependência entre as variáveis ao longo da execução do algoritmo.

Outro aspecto que os CEDAs não têm explorado adequadamente diz respeito às distribuições marginais. Os referidos algoritmos não realizam uma verificação de qual tipo de distribuição marginal é mais adequada na descrição do comportamento das variáveis. Ao definir como parâmetro esta importante parte do modelo, o EDA-MEC mostrou empiricamente que a mudança do tipo de distribuição marginal pode determinar a convergência do algoritmo para o ótimo global.

O algoritmo do modelo proposto deixa como contribuição uma heurística adaptativa de reinicialização que, além de resolver problema de indeterminação da

correlação entre as variáveis, permite ao EDA-MEC inserir a diversidade necessária para sair de ótimos locais. Este mecanismo fornece dados para a construção de uma nova distribuição de busca. Uma parte desses dados provém de variáveis próximas da melhor solução, no momento da reinicialização, e de outras espalhadas aleatoriamente dentro do espaço de busca. Outro mecanismo de diversidade utilizado no EDA-MEC consiste na utilização de uma variação do conceito de indivíduos rebeldes (DELAOSSA *et al.*, 2009), que são obtidos a partir de uma modificação do modelo probabilístico original, com a finalidade de explorar outras regiões do espaço de busca. Empiricamente, esta tese também mostrou que a inclusão de pelo menos um indivíduo rebelde na população pode contribuir, principalmente nas funções mais difíceis, para o aumento de desempenho do EDA-MEC.

Os ensaios realizados com algumas funções *benchmark* clássicas de otimização numérica permitem avaliar positivamente o EDA-MEC. O mesmo obteve desempenho superior ao ser comparado com outros algoritmos de estimação de distribuição baseados em cópulas, em termos de taxa de sucesso e em número de avaliações. Inclusive, obteve desempenho ótimo na função de Rosenbrock, na qual apenas alguns algoritmos, como o CMA-ES, conseguem ser bem sucedidos na localização do mínimo global da mesma.

Apesar de bem sucedido nesses ensaios, até mesmo com 50 e 100 variáveis, o desempenho do algoritmo em problemas de grande escala ainda precisa ser verificado, pois as competições mais recentes de otimização global têm utilizado combinações de funções *benchmark* clássicas com mais de 1.000 variáveis⁷. Além disso, outras propriedades precisam ser verificadas no modelo, incluindo a invariância a transformações no espaço de busca. Isto envolve a utilização de funções *benchmarks* deslocadas e rotacionadas, tais como as definidas nas competições de computação evolutiva.

A continuação do trabalho descrito nesta tese inclui:

- Investigação mais detalhada da complexidade computacional do algoritmo, buscando identificar quais problemas podem ser resolvidos pelo mesmo, as motivações que o levam a falhar, além de estabelecer métricas de graus de dificuldade;

⁷ <http://goanna.cs.rmit.edu.au/~xiaodong/cec13-lsgo/competition/>

- Investigação mais detalhada do mecanismo de funcionamento dos indivíduos rebeldes e sua relação com o tamanho da população. Também é oportuno adotar outros critérios de geração de rebeldes, uma vez que estes não necessariamente têm que ser obtidos do complemento/extremo das amostras (método utilizado nesta tese), podendo ser definidos para explorar o espaço de busca de modo mais gradual;
- Implementação do mais refinado método de seleção dos algoritmos genéticos, a seleção por torneio;
- Ampliação do modelo para outras cópulas e distribuições marginais, bem como a investigação das características que tornam uma marginal adequada para uma cópula em um problema de otimização;
- Emprego de técnicas de redução de variância no processo de simulação de amostras, incluindo a amostragem descritiva (SALIBY, 1989, 1990). Estas técnicas têm apresentado melhores resultados que a amostragem aleatória simples e, no caso do presente trabalho, podem contribuir para realizar uma estimação mais adequada da distribuição, uma vez que constituem métodos amostrais mais controlados, garantindo a varredura efetiva do espaço de busca, sem um aumento correspondente do esforço de amostragem;
- Aplicação de métodos estatísticos para identificar a melhor cópula em cada problema de otimização. Entre esses métodos estão os testes de qualidade de ajuste aos dados (*goodness-of-fit tests*) (GENEST *et al.*, 2009) e o Critério de Informação de Akaike (AIC). Este último é utilizado como método de seleção de cópulas em SALINAS-GUTIÉRREZ *et al.* (2014) e BRECHMANN *et al.* (2012) asseveram que o mesmo é mais confiável que testes de qualidade de ajuste aos dados;
- Desenvolvimento de um sistema inteligente de seleção de parâmetros e de cópulas para otimizar o funcionamento do EDA-MEC;
- Adaptação do modelo para tratar problemas de programação inteira não linear mista;
- Extensão do modelo para permitir o tratamento de problemas práticos de otimização, tipicamente multiobjetivos e/ou dinâmicos;

- Desenvolvimento de novos EDAs, considerando cópulas paramétricas com modelos de mistura de distribuições, particularmente adequados em problemas multimodais.

Paralelamente à investigação do mecanismo de funcionamento dos indivíduos rebeldes, cabe testar uma variação do EDA-MEC relacionada a eles. Como o EDA-MEC descarta os piores indivíduos ao longo do processo evolutivo, as populações de soluções tendem a ficar mais densas em regiões onde ótimos locais foram encontrados. Deste modo, a capacidade do modelo em descrever o espaço de busca, que está diretamente relacionada à densidade de pontos, tende a ser melhor nestas regiões de ótimos locais. A fim de melhorar a habilidade do modelo em descrever o espaço de busca de forma mais homogênea (e assim ter mais chances de encontrar a região do ótimo global), propõe-se dois ajustes:

- i. fazer o algoritmo acumular a população ao longo do processo evolutivo a fim de que o modelo probabilístico não deixe de representar regiões do espaço que possam ser úteis adiante;
- ii. construir o modelo probabilístico usando-se a aptidão dos indivíduos como a função densidade de probabilidade, ao invés de utilizar a densidade espacial das soluções

Referências bibliográficas

AAS, K.; CAZADO, C.; FRIGESSI, A.; BAKKEN, H. Pair-copula constructions of multiple dependence. *Insurance Mathematics and Economics*. Vol.44, no 2, p.182–198, 2007.

ABRAHAM, A.; NEDJAH, N.; MOURELLE, L. M. Evolutionary Computation: from Genetic Algorithms to Genetic Programming. In: NEDJAH, N.; ABRAHAM, A.; MOURELLE, L. (eds.) *Genetic Systems Programming Theory and Experiences*. Studies in Computational Intelligence (SCI), Vol. 13, p.1-20. Germany: Springer, 2006.

BALAKRISHNAN, V.; BOYD, S. Global Optimization in Control System Analysis and Design. In: LEONDRES, C.T. *Control and Dynamic Systems: Advances in Theory and Applications*. New York: Academic Press, 1992. Vol.53, p. 1-56.

BALUJA, S.: Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. *Tech. Rep. CMU-CS-94-163*, Carnegie-Mellon University, 1994.

BALUJA, S.; CARUANA, R. Removing the genetics from the standard genetic Algorithm. In: PRIEDITIS, A.; RUSSEL, S. (Eds.), *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kauffman, Madison, WI, p. 38–46, 1995.

BALUJA, S.; DAVIES, S. Using optimal dependency-tree for combinatorial optimization: Learning the structure of the search space. In: *Proceedings of 14th International Conference on Machine Learning*, p. 30-38. Morgan Kaufmann, 1997.

BEDFORD, T.; COOKE, R. M. Vines—a new graphical model for dependent random variables. *Annals of Statistics*, Vol.30, no 4, p.1031–1068, 2002.

BIANCHI, L.; DORIGO, M.; GAMBARDELLA, L. M.; GUTJAHR, W.J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, Vol. 8, no 2, p. 239–287, 1989.

BIEGLER, L.T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Philadelphia: SIAM, 2010.

BOOKER, L.B.; GOLDBERG, D.E.; HOLLAND, J.H. Classifier Systems and Genetic Algorithms. *Artificial Intelligence*, v. 40, p. 235-282, 1989.

BORKOWF, C. Computing the nonnull asymptotic variance and the asymptotic relative efficiency of Spearman's rank correlation. *Comput. Statist. Data Anal.*, no 39, p. 271–286, 2002.

BOSMAN, P.A.N.; THIERENS, D. Advancing Continuous IDEAs with Mixture Distributions and Factorization Selection Metrics. In: Pelikan, M.; Sastry, K. (organisers), *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference - GECCO-2001*, p. 208-212, 2001.

BOSMAN, P.A.N.; DE JONG, E.D. Combining Gradient Techniques for Numerical Multi-Objective Evolutionary Optimization. In: Keijzer, M. *et al.* (editors), *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2006*, p. 627-634, ACM Press, New York, New York, 2006.

BOSMAN, P.A.N.; GRAHL, J.; THIERENS, D. Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift. In: RUDOLPH, G. *et al.* (editors), *Parallel Problem Solving from Nature - PPSN X*, p. 133-143, Springer-Verlag, Berlin, 2008.

BOSMAN, P.A.N. On Empirical Memory Design, Faster Selection of Bayesian Factorizations and Parameter-Free Gaussian EDAs. In: Raidl, G. *et al.* (editors), *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2009*, p. 389-396, ACM Press, New York, New York, 2009.

BOSMAN, P.A.N. The Anticipated Mean Shift and Cluster Registration in Mixture-based EDAs for Multi-Objective Optimization. In: Branke, J. *et al.* (editors), *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2010*, p. 351-358, ACM Press, New York, New York, 2010.

BRECHMANN, E.C; CZADO, C.; AAS, K. Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, Vol., 40, no 1, p.68–85, 2012.

CASTRO, P. A. D.; VON ZUBEN, F. J. Algoritmos de Estimaco de Distribuico. In: Antnio Gaspar-Cunha; Ricardo Takahashi; Carlos Henggeler Antunes. (Org.). Manual de Computaco Evolutiva e Metaheurstica. 1ed. Coimbra: Imprensa da Universidade de Coimbra, 2012, Vol. 1, p. 213-234.

CHERUBINI, U.; LUCIANO, E.; VECCHIATO, W. *Copula Methods in Finance*. 1st ed. Chichester: John Wiley & Sons Ltd, 2004.

CHOU, H.; PREMKUMAR, G.; CHU, C. Genetic algorithms for communications network design - an empirical study of the factors that influence performance. *IEEE Transactions on Evolutionary Computation*, Vol.5, no 3, p.236-249, Jun. 2001.

CUESTA-INFANTE, A.; SANTANA, R.; HIDALGO, J.I.; BIELZA, C.; LARRAÑAGA, P. Bivariate empirical and n-variate Archimedean copulas in estimation of distribution algorithms. In: *Proceedings of the 2010 Congress on Evolutionary Computation (CEC-2010)*, 2010.

DANTZIG, G. B.; FULKERSON, R.; JOHNSON, S.M. Solution of a large-scale traveling salesman problem. *Operations Research*, Vol.2, p.393-410, 1954.

DARWIN, C. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. London: John Murray, 1859.

DAVIS, L. *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.

DE BONET, J.S; ISBEL, C.L; VIOLA, P. MIMIC: Finding optima by estimating probability densities. In: *Advances in neural information processing systems*, Vol. 4, p.424-430. The MIT Press, 1997.

DEHEUVELS, P. La fonction de dépendance empirique et ses propriétés – Un test non paramétrique d'indépendance, Académie Royale de Belgique – Bulletin de la Classe des Sciences – 5e Series, no 65, p.274-292, 1979.

DELAOSSA, L.; GÁMEZ, J.A.; MATEO, J.L.; PUERTA, J.M. Avoiding premature convergence in Estimation of Distribution Algorithms. In: *IEEE Congress on Evolutionary Computation (CEC 2009)*, p. 455–462, 2009.

DEMPSTER, A.P.; LAIRD, N.M.; RUBIN, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol.39, no 1, p.1–38, 1977.

DONG, W.; CHEN, T.; TINO, P.; YAO, X. Scaling Up Estimation of Distribution Algorithms for Continuous Optimization. *IEEE Transactions on Evolutionary Computation*, Vol.17, no 6, p.797-822, 2013.

EMBRECHTS, P.; MCNEIL, A.J.; STRAUMANN D. Correlation and Dependence in Risk Management: Properties and Pitfalls, Preprint ETH Zurich, 1999. Online available at: < <ftp://ftp.math.ethz.ch/users/embrecht/pitfalls.pdf> >. Accessed: Jul, 2013.

EMBRECHTS, P.; LINDSKOG, F e MCNEIL, A. Modelling dependence with copulas and applications to risk management. In: RACHEV, S.T. *Handbook of heavy tailed distributions in finance*. Amsterdam: Elsevier/North-Holland, 2003.

FOGEL, L.J.; OWENS, A.J.; WALSH, M.J. *Artificial Intelligence through Simulated Evolution*. New York: John Wiley, 1966.

FOUSKAKIS, D.; DRAPER, D. Stochastic optimization: a review. *International Statistical Review*, Vol.70, no 3, p. 315-349, Dec. 2002.

FREES, E.W.; VALDEZ, E. Understanding relationship using copulas. *N. Amer. Actuarial J.*, No 2, p. 1–25, 1988.

FRIEDMAN, J. An overview of predictive learning and function approximation. *NATO ASI Series of Computer and Systems Sciences*, Vol. 136, p. 1–61, 1994.

GAO, Y. Multivariate Estimation of Distribution Algorithm with Laplace Transform Archimedean Copula. In: HU, W.; LI, X. (eds.) *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS 2009)*, 2009.

GAO, Y.; PENG, L.; LI, F.; LIU, M.; HU, X. Multiobjective Estimation of Distribution Algorithms Using Multivariate Archimedean Copulas and Average Ranking. In: WEN, Z.; LI, T. (editors), *Foundations of Intelligent Systems. Advances in Intelligent Systems and Computing*, Vol. 277, p.591-601. Berlin: Springer, 2014.

GEEM, Z.; KIM, J.; LOGANATHAN, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*. Vol. 76, no 2, p.60-68, 2001.

GENEST, C.; RIVEST, L.-P. Inference procedures for bivariate archimedean copulas. *Journal of the American Statistical Association*, no 88, p.1034-1043, 1993.

GENEST, C.; GHOUDI, K.; RIVEST, L.-P. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, no 82, p.543–552, 1995.

GENEST, C., RÉMILLARD, B.; BEAUDOIN, D. Goodness-of-fit tests for copulas: A review and a power study, *Insurance: Mathematics and Economics*, 44, 199–213, 2009.

GENEST, C.; CARABARÍN-AGUIRRE, A.; HARVEY, F. Copula parameter estimation using Blomqvist's beta. *Journal de la Société Française de Statistique*, Vol.154, no 1, p.5-24, 2013.

GEORGIEVA, A.; JORDANOV, I. Global optimization based on novel heuristics, low-discrepancy sequences and genetic algorithms. *European Journal of Operational Research*, no 196, p.413-422, Jan. 2009.

GOLDBERG, D.E. *Genetic algorithms in search, optimization and machine learning*. Boston: Addison-Wesley Longman Publishing Co., Inc., 1989.

GONÇALVES, A.R.; VON ZUBEN, F.J. Online learning in estimation of distribution algorithm for dynamic environments. In: *Proceedings of IEEE Congress on Evolutionary Computation, CEC'11*, New Orleans, 2011.

GONZÁLEZ-FERNÁNDEZ, Y. *Algoritmos con estimación de distribuciones basados en cópulas y vines*. Bachelor thesis, University of Havana, Cuba, 2011.

GONZÁLEZ-FERNÁNDEZ, Y.; SOTO, M. *copulaedas*: An R package for estimation of distribution algorithms based on copulas, 2012. <<http://cran.open-source-solution.org/web/packages/copulaedas/vignettes/copulaedas-manual.pdf>>. Accessed: Jun 2013.

GONZÁLEZ-FERNÁNDEZ, Y.; SOTO, M. *copulaedas*: An R Package for Estimation of Distribution Algorithms Based on Copulas, 2014. *Journal of Statistical Software*, Vol. 58, no 9, p. 1-34. <<http://www.jstatsoft.org/v58/i09/>>. Accessed: Jul 2014.

GRANELLI, G.P.; MONTAGNA, M. Genetic algorithm applications to the solution of electric power system problems. In: LEFEBVRE, C. M. *Electric power: generation, transmission and efficiency*. New York: Nova Science Publishers, 2007.

GUO, X.; WANG, L.; ZENG, J.; ZHANG, X. VQ Codebook Design Algorithm Based on Copula Estimation of Distribution Algorithm, *First International Conference on Robot, Vision and Signal Processing*, p.178-181, 2011.

HANSEN, N.; OSTERMEIER, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, Vol.9, no 2, p.159-195, 2001.

HANSEN, N.; ROS, R.; MAUNY, N.; SCHOENAUER, M.; AUGER, A. Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, Vol.11, p.5755–5769, 2011.

HARIK, G.R., CANTÚ-PAZ, E., GOLDBERG, D., MILLER, B.: The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evol. Comput.* Vol.7, no 3, p.231–253, 1999.

HARIK, G.R.; LOBO, F.G.; SASTRY, K. Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In: PELIKAN, M., SASTRY, K.; CANTÚ-PAZ, E. *Scalable Optimization via probabilistic Modeling*, Springer, p.39-61, 2006.

HAUSCHILD, M; PELIKAN, M. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, Vol. 1, no 3, p. 111-128, Sep. 2011.

HOLLAND, J. H. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1975.

HYRŠ, M.; SCHWARZ, J. Multivariate Gaussian copula in estimation of distribution algorithm with model migration. In: *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, p.114-119, 9-12, Dec. 2014.

JOE, H.; XU, J. *The estimation method of inference function for margins for multivariate models*. Technical Report no 166. Vancouver: Univ. of British Columbia, Dept. of Statistics, 1996.

JOE, H. *Multivariate Models and Dependence Concepts*. London: Chapman and Hall, 1997.

KOZA, J.R. *Genetic Programming: On the Programming of Computers by means of Natural Selection*. Cambridge: The MIT Press, 1992.

KUROWICKA, D.; COOKE, R. M. Conditional, partial, and rank correlation for the elliptical copula; Dependence modeling in uncertainty analysis. European Safety and Reliability Conference, ESREL 2001. In: ZIO, E.; DEMICHELA, M.; PICCININI, N. *Safety and Reliability*, Vol.3, p.1795-1802, 2001.

LANDSMAN, Z.M.; VALDEZ, E.A. Tail conditional expectations for elliptical distributions. *N. Am. Actuar. J.*, Vol.7, p. 55–71, 2003.

LARRAÑAGA, P.; ETXEBERRIA, R.; LOZANO, J.; PENA, J. Optimization by learning and simulation of Bayesian and Gaussian networks. *Tech. Rep. EHU-KZAAIK-IK-4/99*, Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.

LARRAÑAGA, P. A Review of Estimation of Distribution Algorithms. In: LARRAÑAGA, P.; LOZANO, J.A. (editors) *Estimation of Distribution Algorithms: A new tool for Evolutionary Computation*. Boston: Kluwer Academic Publishers, 2001.

LARRAÑAGA, P.; LOZANO, J.A. *Estimation of distribution algorithm: a new tool for evolutionary computation*. Boston: Kluwer Academic Publishers, 2002.

LING, C.H. Representation of associative functions. *Publication Mathematicae Debrecen*, no 12, p. 189-212, 1965.

LOZANO, J.; LARRAÑAGA, P.; INZA, I.; BENGOETXEA, E. (eds.). *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Studies in Fuzziness and Soft Computing, Vol. 192. Berlin: Springer, 2006.

LOZANO, M.; HERRERA, F.; CANO, J. R. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences*, Vol. 23, no 178, p. 4421–4433, 2008.

LUENBERGER, D; YINYU, Y. *Linear and nonlinear programming*. International Series in Operations Research & Management Science. 3rd ed. New York: Springer, 2008.

MAHFOUD, S.W. *Niching methods for genetic algorithms*. Illinois Genetic Algorithms Laboratory (IlligAL) - University of Illinois at Urbana-Champaign, Tech. Rep. 95001, 1995.

MAI, J-F.; SCHERER, M. *Simulating Copulas: Stochastic Models, Sampling Algorithms and Applications*. Series in Quantitative Finance – v.4. London: Imperial College Press, 2012.

MARSHALL, A.W.; OLKIN, I. Families of multivariate distributions, *J. Amer. Statist. Assoc.*, no 83, p. 834–841, 1988.

MCNEIL, A. J. Sampling nested Archimedean copulas. *Journal of Statistical Computation and Simulation*. Vol. 78, no 6, p.567–581, 2008.

MENDES, B.V.M. *Introdução à análise de eventos extremos*. Rio de Janeiro: E-papers Serviços Editoriais, 2004.

MENDES, B.V.M.; SEMERARO, M.M.; LEAL, R.C. Pair-copulas modeling in finance. *Tech. rep.*, IM/COPPEAD, Federal University at Rio de Janeiro, Brazil, 2009.

MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. 2nd (extended) ed. New York: Springer, 1994.

MICHALEWICZ, Z.; FOGEL, D.B. *How to solve it: modern heuristics*. New York: Springer, 2000.

MOSCATO, P. Memetic algorithms. In: *Handbook of Applied Optimization*. Oxford University Press, 2002. Chap. 3.6.4, p.157-167.

MOSCATO, A. Complete Search in Continuous Global Optimization and Constraint Satisfaction. In: ISERLES, A. *Acta Numerica 2004*, Cambridge University Press, 2004, p. 271–369.

MÜHLENBEIN, H.; PAAß, G. From recombination of genes to the estimation of distributions I. Binary parameters. In: VOIGT, H.-M *et al.* (eds.). *Parallel Problem Solving from Nature IV: Lecture Notes in Computer Sciences*, Vol. 1141, p.178–187, 1996.

NELSEN, R. B. *An introduction to copula*. 2nd ed. New York: Springer, 2006.

OAKES, D. Multivariate survival distributions. *J. Nonparametr. Statist.*, no 3, p.343–354, 1994.

ORTIZ-BOYER, D.; HERVÁS-MARTÍNEZ, C.; GARCÍA-PEDRAJAS, N. CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features. *Journal of Artificial Intelligence Research*, no 24, p.1–48, 2005.

PELIKAN, M.; MÜHLENBEIN, H. Marginal distributions in evolutionary algorithms. In: *Proceedings of the International Conference on Genetic Algorithms Mendel'98*, p.90-95, 1999a.

PELIKAN, M.; GOLDBERG, D.E.; CANTÚ-PAZ, E. BOA: The Bayesian Optimization Algorithm. In: BANZHAF, W.; DAIDA, J.; EIBEN, A. E.; GARZON, M.H.; HONAVAR, V.; JAKIELA, M.; SMITH, R.E. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, v. I. San Francisco: Morgan Kaufmann Publishers, 1999b. p. 525–532.

PELIKAN, M.; GOLDBERG, D.E.; CANTÚ-PAZ, E. Linkage problem, distribution estimation and Bayesian networks. *Evolutionary Computation*, Vol. 8, no 3, p.311–340, 2000.

PELIKAN, M. *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. 1st edition. Studies in Fuzziness and Soft Computing, Vol. 170. Berlin: Springer, 2005.

PENA, J.M.; LOZANO, J.A.; LARRAÑAGA, P. Benefits of data clustering in multimodal function optimization via EDAs. In: LARRAÑAGA, P., LOZANO, J.A. *Estimation of Distribution Algorithms: A new tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

PINTÉR, J.D. Global optimization: software, test problems and applications. In: PARDALOS, P.M.; ROMEINJN, H.E. *Handbook of global optimization: volume 2*. Dordrecht: Kluwer Academic Publishers, 2002. Cap.15, p.515–569.

RECHENBERG, I. Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment*, Library Translation no 1122, 1965.

RECHENBERG, I. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag, 1973.

REDDY, M. J.; GANGULI, P. Bivariate Flood Frequency Analysis of Upper Godavari River Flows Using Archimedean Copulas. *J Water Resources Management*, Vol. 26, no 14, p. 3995-4018, 2012.

SALIBY, E. *Repensando a Simulação: A Amostragem Descritiva*. São Paulo: Ed. Atlas, 1989.

SALIBY, E. Descriptive Sampling: A Better Approach to Monte Carlo Simulation. *Journal of the Operational Research Society*, Vol. 41, no 12, p.1133-1142, 1990.

SALINAS-GUTIÉRREZ, R.; HERNÁNDEZ-AGUIRRE, A.; VILLA-DIHARCE, E. Using copulas in estimation of distribution algorithms. In: *Advances in Artificial Intelligence (MICAI'09). Lecture Notes in Computer Science*, Vol. 5845, p. 658–668. Berlin: Springer, 2009.

SALINAS-GUTIÉRREZ, R.; HERNÁNDEZ-AGUIRRE, A.; VILLA-DIHARCE, E. D-vine EDA: A New Estimation of Distribution Algorithm Based on Regular Vines. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*, p. 359–365, 2010

SALINAS-GUTIÉRREZ, R.; HERNÁNDEZ-AGUIRRE, A.; VILLA-DIHARCE, E. Dependence Trees with Copula Selection for Continuous Estimation of Distribution Algorithms. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, p.585–592, 2011.

SALINAS-GUTIÉRREZ, R.; HERNÁNDEZ-AGUIRRE, A.; VILLA-DIHARCE, E.R. Copula selection for graphical models in continuous Estimation of Distribution Algorithms. *Computational Statistics*, Vol. 29, no 3-4, p.685–713, 2014.

SANTANA, R. Estimation of Distribution Algorithms with Kikuchi approximations, *Evolutionary Computation*, Vol.13, no 1, p.67-97, 2005.

SAVU, C.; TREDE, M. Hierarchical Archimedean copulas. *International Conference on High Frequency Finance Konstanz Germany*, p. 1–17, 2006.

SCHWARZ, G. Estimating the dimension of a model. *Annals of Statistics*, Vol. 6, p. 461–464, 1978.

SCHWEFEL, H.-P. *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.

SONG, X.; TANG, L. A novel hybrid Differential Evolution-Estimation of Distribution Algorithm for dynamic optimization problem. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2013)*, p.1710–1717, 2013.

SOTO, M.; OCHOA, A.; ARDERÍ, R.J. Gaussian Copula Estimation of Distribution Algorithm. *Technical Report ICIMAF 2007-406*, Institute of Cybernetics, Mathematics and Physics, Cuba, 2007.

SOTO, M.; GONZÁLEZ-FERNÁNDEZ, Y. Vine Estimation of Distribution Algorithms. *Technical Report ICIMAF 2010-561*, Institute of Cybernetics, Mathematics and Physics, Cuba, 2010

STORN, R.; PRICE, K. Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, Kluwer Academic Publishers, Vol. 11, p. 341-359, 1997.

SKLAR, A. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, Vol. 8, p.229–231, 1959.

SUGANTHAN, P. N. ; HANSEN, N. ; LIANG, J. J. ; DEB, K. ; CHEN, Y.-P. Chen ; AUGER, A.; TIWARI, S. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, May 2005. <http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm>.

Accessed: Jun 2013.

SUN, J.; GARIBALDI, J.M.; HODGMAN, C. Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 9, no 1, p.185-202, 2012.

SYSWERDA, G. Uniform Crossover in Genetic Algorithms. In: SCHAFFER, J.D. (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, p. 2-9, 1989.

TALBI, E-G. *Metaheuristics: from design to implementation*. New Jersey: Wiley, 2009.

THIERENS, D.; BOSMAN, P.A.N. Multi-objective mixture-based iterated density estimation evolutionary algorithms. *Morgan Kaufmann*, p. 663–670, 2001.

VANDERPLAATS, G. N. *Numerical optimization techniques for engineering design: with applications*. New York: McGraw-Hill, 1984.

VON ZUBEN, F. J. Computação Evolutiva: Uma Abordagem Pragmática. Anais da I Jornada de Estudos em Computação de Piracicaba e Região (1a JECOMP), Vol. 1, p. 25-45, 2000.

VON ZUBEN, F.J. Notas de aula da disciplina IA707 - Computação Evolutiva, Programa de Pós-Graduação, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2011.

WALLIN, D.; RYAN, C. On the diversity of diversity. In: *IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, p. 95–102.

WANG, L., ZENG, J; HONG. Y. Estimation of Distribution Algorithm Based on Copula Theory. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009)*. 2009a. p.1057–1063.

WANG, L.F.; ZENG, J.C.; HONG, Y. Estimation of distribution algorithm based on Archimedean copulas. In: *First ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC'09)*. 2009b. p. 993–996.

WANG, L.F.; ZENG, J.C. Estimation of distribution algorithm based on copula theory. In: *Exploitation of Linkage Learning in Evolutionary Algorithms. Evolutionary Learning and Optimization*, Vol. 3, p. 139–162. Berlin: Springer, 2010.

WEISE, T. *Global Optimization Algorithms – Theory and Application*. Germany: self-published, 2009. Online available at: <<http://it-weise.de/projects/book.pdf>>. Accessed: Jul, 2013.

WHELAN, N. Sampling from Archimedean copulas. *Quantitative Finance* 4, p.339–352, 2004.

WOLPERT, D. H.; MACREADY, W.G. No free-lunch theorems for search. *Tech. rep. 95-02-010*, Santa Fe Institute, 1995.

YE, B.; GAO, H.; WANG, X.; ZENG, J. Estimation of Distribution Algorithm Based on Nested Archimedean Copulas Constructed with Lévy Subordinators. In: *Proceedings of the Eleventh International Conference on Computer-Aided Industrial Design & Conceptual Design (CAIDCD 2010)*, p.1586–1590, 2010.

ZABINSKY, Z.B. *Stochastic Adaptive Search for Global Optimization*. Boston: Kluwer Academic Publishers, 2003.

ZURADA, J.M.; MARKS II, R.J.; ROBINSON, C.J. (Eds.) *Computational Intelligence – Imitating Life*, IEEE Press, 1994.