

**OPERADORES DE BUSCA LOCAL BASEADA EM
APROXIMAÇÃO LINEAR-QUADRÁTICA PARA
OTIMIZAÇÃO DE FUNÇÕES RUIDOSAS**

ANDRÉ RODRIGUES DA CRUZ

**OPERADORES DE BUSCA LOCAL BASEADA EM
APROXIMAÇÃO LINEAR-QUADRÁTICA PARA
OTIMIZAÇÃO DE FUNÇÕES RUIDOSAS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, Escola de Engenharia como requisito parcial para a obtenção do grau de Doutor em Engenharia Elétrica.

ORIENTADOR: RICARDO HIROSHI CALDEIRA TAKAHASHI

17 de outubro de 2017

© 2017, André Rodrigues da Cruz.
Todos os direitos reservados.

da Cruz, André Rodrigues

C957o Operadores de busca local baseada em aproximação
linear-quadrática para otimização de funções ruidosas /
André Rodrigues da Cruz. — , 2017
xxx, 157 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas
Gerais, Escola de Engenharia
Orientador: Ricardo Hiroshi Caldeira Takahashi

1. Engenharia elétrica – Teses. 2. Algoritmos
evolutivos – Teses. 3. Vacinação – Teses. 4. Heurística
— Teses. I. Takahashi, Ricardo Hiroshi Caldeira. II.
Universidade Federal de Minas Gerais. Escola de
Engenharia. III. Título.

CDU 621.3(043)

**"Operadores de Busca Local Baseada em
Aproximação Linear-quadrática para
Otimização de Funções Ruidosas"**

André Rodrigues da Cruz

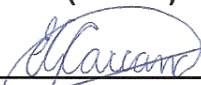
Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 17 de outubro de 2017.

Por:



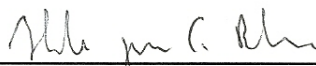
**Prof. Dr. Ricardo Hiroshi Caldeira Takahashi
DMAT (UFMG) - Orientador**



**Prof. Dr. Eduardo Gontijo Carrano
DEE (UFMG)**



**Prof. Dr. Luiz Henrique Duczmal
Dep. Estatística (UFMG)**



**Prof. Dr. Hélio José Correa Barbosa
(LNCC)**



**Prof. Dr. Alexandre Cláudio Botazzo Delbem
Dep. de Sistemas de Computação (USP)**

Para Cecília Lopes da Silva.

Agradecimentos

“If I have seen further it is by standing on the shoulders of Giants.” escreveria Isaac Newton. Eu, André Rodrigues da Cruz, se cheguei até aqui... é porque tive o incentivo, o apoio e o carinho de muitas pessoas especiais para mim. Não há palavras em meu vocabulário para descrever a imensidão do quanto sou grato a elas.

Inicialmente registro aqui gratidão a Deus, o criador, o ser onipresente e onipotente que permite o livre-arbítrio. Questiono sobre estar entre um mero acaso, ou alguma razão especial, ou uma combinação linear destes fatores para que eu tenha chegado aqui... ou nenhuma das opções anteriores. Fato é que cheguei, tive a oportunidade e os desafios que ela impôs. Sou muito agradecido pela conjuntura favorável de tempo, espaço e seres humanos maravilhosos ao qual fui exposto para que me inspirasse e espelhasse. Devaneio sobre o que está por vir, sobre as responsabilidades e alegrias que terei com esta conquista.

Agradeço a minha família. Aos meus amados pais Olga Maria da Cruz e Eustáquio Rodrigues da Cruz, que sempre me proveram o melhor deles, o mais belo e puro amor, a leveza de superar os desafios do dia-a-dia e um caminho digno para seguir na vida. Agradeço aos meus estimados irmãos Nilton Célio Rodrigues Cruz, Nilcemar Rodrigues Carvalho Cruz e Aleksandro Rodrigues da Cruz, pelo imprescindível apoio que tive para que eu atingisse tal esse triunfo.

Agradeço a minha linda Cecília Lopes da Silva, a mulher espetacular que sempre me incentivou e torceu para que essa vitória ocorresse. Ao meu lado, ela acompanhou toda a trajetória, do início ao fim, minhas empolgações, decepções, dificuldades e superações. Transcender foi uma tarefa muito árdua, e o carinho dela foi essencial para que forças surgissem como sequências de pulsos em meu ser.

Agradeço ao meu querido orientador Ricardo Hiroshi Caldeira Takahashi, de quem desde 2005 recebo valiosos conhecimentos, não apenas de cunho técnico-científico, mas também como exemplo de maestria e humanidade a ser seguido. Toquemos o barco então Taka, para os próximos projetos!

Agradeço a todos os meus professores que, desde a infância, contribuíram com

minha formação. Carrego em mim a dedicação e a paixão pela docência que vi e admirei neles. Um agradecimento especial aos queridos professores e amigos que fiz dentro do Programa de Pós-Graduação em Engenharia Elétrica da UFMG, Alexandre Celestino Leite Almeida, Eduardo Gontijo Carrano, Elizabeth Fialho Wanner, Luiz Henrique Duczmal, Oriane Magela Neto (*in memoriam*), Rodrigo Tomas Nogueira Cardoso, Rodney Rezende Saldanha e Sérgio Augusto Araújo da Gama Cerqueira.

Agradeço a todos os meus amigos, dos mais distintos locais e épocas de minha vida, que torceram por mim. Agradeço também aos meus estimados familiares pelo carinho. Obrigado a todos pela força, incentivo e compreensão.

Resumo

Esta tese apresenta um estudo a respeito do problema de otimização de funções ruidosas por meio de heurísticas da classe dos algoritmos evolutivos. Mais especificamente, esta tese propõe a utilização de um tipo de operador de busca local, baseado em aproximações lineares-quadráticas, como mecanismo para o tratamento dos ruídos presentes na classe das funções ruidosas. A hipótese subjacente a tal proposta é a de que a filtragem implícita do ruído que é realizada pelo próprio processo de construção da aproximação de função possa causar um efeito de atenuação do efeito do ruído no comportamento do algoritmo, assim favorecendo seu desempenho na busca do ótimo. Essa abordagem é diferente daquela que vem sendo majoritariamente empregada na literatura para o tratamento de funções ruidosas, e que normalmente se baseia na reamostragem de avaliações de soluções visando o aumento da confiança nas operações de comparação realizadas pelo algoritmo.

Para a realização de experimentos numéricos, foi empregado como algoritmo-base a Estratégia Evolutiva $(1 + 1)$ -ES, que foi denominado ES-BS. Esse algoritmo-base foi modificado das seguintes maneiras: (i) foi incluído um operador de busca local baseado em busca em linha do tipo Quasi-Newton, sendo o algoritmo resultante denominado ES-QN; (ii) foi incluído um operador de busca local baseado em aproximações lineares-quadráticas de funções, resultando no algoritmo ES-AP; (iii) foi incluído um operador de reamostragem de soluções baseado em estimativas de confiança, resultando no algoritmo ES-CC; e (iv) foi feita a inclusão tanto do operador de busca local por aproximação de funções quanto do operador de reamostragem, que resultou no algoritmo ES-APCC. Os algoritmos foram então submetidos a três conjuntos de experimentos.

Primeiro, apenas os algoritmos sem reamostragem foram avaliados, sobre um conjunto clássico de funções *benchmark* não ruidosas. Nesse experimento, o algoritmo ES-AP se mostrou superior aos algoritmos ES-BS e ES-QN, indicando que o operador de busca local por aproximação de funções funciona adequadamente mesmo fora do contexto de funções ruidosas. A seguir, as funções utilizadas no experimento anterior foram convertidas em funções ruidosas, pela adição de ruído Gaussiano, sendo então

realizada a avaliação do desempenho dos cinco algoritmos. O algoritmo ES-AP demonstrou novamente um desempenho melhor que todos os demais, sendo ainda observado que nas únicas funções para as quais o ES-AP não atingiu o primeiro lugar, o algoritmo situado em primeiro foi o ES-APCC. Finalmente, uma função originalmente ruidosa foi considerada. Essa função descreve o efeito de campanhas de vacinação em populações, sendo essas campanhas realizadas com o objetivo de controlar a propagação de epidemias. A função representa um custo da implementação do controle em relação a modelagem do fenômeno feita a partir da simulação de interações probabilísticas entre indivíduos, o que a torna ruidosa por natureza. Todos os cinco algoritmos foram avaliados na tarefa de otimização dessas políticas de vacinação. Neste experimento, o algoritmo de melhor desempenho foi o ES-APCC, seguido do ES-CC.

Uma análise geral dos experimentos realizados fornece suporte à hipótese principal examinada nesta tese, de que operadores de busca local baseados em aproximação de funções podem vir a constituir mecanismos importantes para a construção de algoritmos orientados para a otimização de funções ruidosas.

Por fim, é importante registrar que duas linhas de investigação relativamente independentes da linha principal deste trabalho tiveram de ser desenvolvidas concomitantemente, no âmbito desta tese, com o objetivo de preencher lacunas metodológicas para as quais se faziam necessários desenvolvimentos suplementares. A primeira destas foi o desenvolvimento de um sistema de sintonização de parâmetros de algoritmos de otimização. Tal sistema se fez necessário como um requisito para permitir a adequada comparação entre algoritmos diferentes, a qual deveria ser realizada sobre classes de distintas funções. O sistema que foi aqui desenvolvido parece ter desempenho que supera aquele exibido por outras heurísticas apresentadas na literatura para executar a mesma função. A segunda foi o desenvolvimento de estudos sobre técnicas de projeto de campanhas de vacinação para o controle de epidemias. Esse estudo se fez necessário para gerar um caso real de função-objetivo ruidosa, a ser empregado na avaliação dos algoritmos propostos nesta tese. Os resultados obtidos nestes estudos também constituem, por si próprios, contribuições originais desta tese, que incluem o desenvolvimento de uma metodologia de simulação estocástica integrada ao processo de projeto da estratégia de controle, o desenvolvimento de uma nova metodologia de projeto da estratégia de controle que articula uma fase transiente e uma fase estacionária do processo de vacinação, e o desenvolvimento de uma metodologia para avaliar a robustez das políticas de vacinação projetadas.

Abstract

This thesis presents an investigation on the problem of noisy function optimization by heuristics of the class of evolutionary algorithms. More specifically, this thesis proposes the employment of a local search operator based on linear-quadratic approximations, as a tool for dealing with noisy functions. The underlying hypothesis is that the implicit filtering that is accomplished by the procedure of function approximation construction may cause an attenuation effect on the noise, in this way enhancing the algorithm performance. This approach is essentially different from the mainstream approach in literature, which is based on objective function re-evaluation procedures that are intended to increase the confidence in comparison operations that are performed within the algorithm.

In order to perform numerical experiments, an Evolutionary Strategy (1 + 1)-ES was employed as the basis algorithm (being denoted by ES-BS). This basis algorithm was modified in the following ways: (i) a local search operator based on a Quasi-Newton line search was included (denoted by ES-QN); (ii) a local search operator based on linear-quadratic function approximation was included (denoted by ES-AP); (iii) a re-sampling operator based on confidence interval estimate was included (denoted by ES-CC); (iv) another algorithm version included both the re-sampling operator and the local search operator based on linear-quadratic function approximation (denoted by ES-APCC). The algorithms were submitted to three experimental procedures.

First, only the algorithms without re-sampling were evaluated, on a classical set of noise-free benchmark functions. In this experiment, the ES-AP presented better performance than the ES-BS and the ES-QN, which suggests that the local search operator based on function approximation works well even outside the context of noisy functions. Next, the noise-free functions that were employed in the former experiment were converted in noisy functions by the adding Gaussian noise to them. The five algorithms were tested on this set of functions, and the ES-AP presented the better general performance. In the few cases in which the ES-AP was not placed first in the performance ranking, the best algorithm was the ES-APCC. Finally, a real noisy func-

tion was considered. This function describes the effect of vaccination campaigns that are intended to control the propagation of an epidemics in a population. The function represents a cost of the implementation of the control in relation to the phenomenon modeling made from the simulation of probabilistic interactions between individuals, which means that this function is noisy by essence. All the five algorithms were evaluated in the task of optimization of the vaccination campaigns, and the best performance was presented by ES-APCC, followed by ES-CC.

The main hypothesis that was examined in this thesis, namely the hypothesis that local search operators based on function approximation might constitute important tools for building algorithms for noisy functions, was corroborated by the experimental results.

Finally, it is important to mention that two research themes relatively independent of the main focus of this work were developed simultaneously, within this thesis, with the aim of filling methodological gaps for which some supplementary developments were necessary. The first one was the development of a parameter tuning system for optimization algorithms. Such a system was necessary as a requisite for allowing suitable comparisons between different algorithms, considering distinct classes of functions. The developed system presents a performance which seems to be better than the performances of different heuristics described in literature for the same purpose. The second one was the development of studies about design techniques of vaccination campaigns for epidemics control. Such a study was necessary in order to give rise to a real noisy objective function which would be employed in the evaluation of the algorithms proposed in this thesis. The results obtained in that studies also constitute original contributions of this thesis, which include: the development of a methodology of stochastic simulation integrated to the control strategy design; the development of a new methodology for the control strategy design which articulates a transient phase and a stationary phase; and the development of a methodology for evaluating the robustness of the resulting vaccination strategies.

Lista de Figuras

3.1	Estratégia Evolutiva (1+1)-ES designada por ES-BS.	21
3.2	(1+1)-ES com busca local baseada em Quasi-Newton designada por ES-QN.	23
3.3	Função <i>QuasiNewton</i> utilizada para encontrar um mínimo local de uma função não linear.	24
3.4	(1+1)-ES com busca local baseada na otimização de aproximação linear-quadrática designada por ES-AP.	31
3.5	Função <i>QuasiNewtonApx</i> de otimização da aproximação linear-quadrática.	33
3.6	(1+1)-ES com operador de comparação com confiança estatística designada por ES-CC.	34
3.7	Função <i>Compara</i> que confronta duas soluções com base no teste-t e realiza reavaliações para reduzir o desvio.	36
3.8	(1+1)-ES com busca local baseada na otimização de aproximação linear-quadrática e comparação com confiança estatística designada por ES-APCC.	38
4.1	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação a todas as funções-objetivo.	44
4.2	Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo obtido pelos algoritmos pela progressão do número de avaliações, em relação a todas as funções-objetivo.	45
4.3	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.	48
4.4	Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo real obtido pelos algoritmos pela progressão do número de avaliações, em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.	49
4.5	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivos reais retornados pelos algoritmos em relação a todas as funções-objetivo em cada nível de ruído.	53

4.6	Beanplot para o $\log(f(x) + 1)$ do custo das melhores políticas de vacinação retornados pelos algoritmos.	62
4.7	Beanplot para o tempo de execução dos algoritmos, em segundos.	64
A.1	Algoritmo <i>Robust Parameter Searcher</i> (RPS) para sintonização de parâmetros de algoritmos de otimização.	97
A.2	Processo de extração da amostra de melhores objetivos induzido pelo parâmetro π_{DE} , obtido pela heurística \mathcal{A}_{mt} , e 100 execuções da DE para cada função-objetivo.	105
A.3	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.	106
A.4	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.	107
A.5	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.	108
A.6	Beanplot para o $\log(f(x) + 1)$ dos valores de objetivo objetivos retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.	109
A.7	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.	110
A.8	Evolução das medianas do $\log(f(x) + 1)$ para o melhor valor de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros para 500, 2000, 3000, 5000 e 10000 avaliações.	111
A.9	Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 231859,8154, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos não ruidosos.	114
A.10	Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 8549,8055, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos não ruidosos.	116
A.11	Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 21,3743, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos não ruidosos.	117

A.12	Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo 252141,1039, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos ruidosos.	118
A.13	Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo 366736,8967, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos ruidosos.	119
A.14	Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo -45864,5972, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos ruidosos.	121
A.15	Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo 404031,8039, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-CC em objetivos ruidosos.	122
A.16	Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo -24167,0778, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-APCC em objetivos ruidosos.	123
B.1	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação à cada função-objetivo no experimento sem ruído.	129
B.2	Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo obtido pelos algoritmos pela progressão do número de avaliações, em relação à cada função-objetivo no experimento sem ruído.	133
B.3	Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.	135
B.4	Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo real obtido pelos algoritmos pela progressão do número de avaliações, em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.	137

Lista de Tabelas

2.1	Trabalhos recentes com aplicações de algoritmos evolutivos na otimização de funções-objetivo ruidosas.	9
2.2	Trabalhos recentes com aplicações de otimização com modelos de aproximação funcional presentes na literatura.	14
2.3	Versões comuns do ES encontradas na literatura.	17
3.1	Organização dos dados para o teste de Quade.	29
4.1	Conjunto de funções-objetivo com os valores estimados de amplitude no valor de objetivo em um raio de distância 1 do ótimo.	43
4.2	Parâmetros sintonizados para objetivos não ruidosos.	43
4.3	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação a todas as funções-objetivo sem ruído.	44
4.4	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação a todas as funções-objetivo sem ruído.	44
4.5	Ranking da mediana do valor da função-objetivo para as diferentes heurísticas, considerando cada função-objetivo, para problemas não-ruidosos. Considera-se empate no caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas heurísticas no ranking 1, a terceira heurística recebe ranking 2 se ficar empatada com uma das duas primeiras, ou ranking 3 se for significativamente diferente das duas primeiras.	46
4.6	Parâmetros sintonizados para objetivos ruidosos.	47
4.7	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.	48
4.8	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.	49

4.9	Ranking da mediana do valor da função-objetivo para as diferentes heurísticas, considerando cada função-objetivo, agregando-se todos os níveis de ruído. Considera-se empate no caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas ou mais heurísticas em determinado ranking, uma heurística situada logo abaixo recebe ranking um a mais se ficar empatada com uma das anteriores, ou ranking igual ao número de heurísticas anteriores mais um se for significativamente diferente das anteriores. A entrada na tabela indicada com (*) indica heurística não equivalente às outras em decorrência da existência de expressivo número de <i>outliers</i>	55
4.10	Estatísticas para o $\log(f(x)+1)$ dos valores de função-objetivo das melhores políticas de vacinação retornadas pelos algoritmos.	63
4.11	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação ao logaritmo do valor de função-objetivo das melhores políticas de vacinação encontradas.	63
4.12	Ranking da mediana do valor da função-objetivo para as diferentes heurísticas. Considera-se empate o caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas ou mais heurísticas em determinado ranking, uma heurística situada logo abaixo recebe ranking um a mais se ficar empatada com uma das anteriores, ou ranking igual ao número de heurísticas anteriores mais um se for significativamente diferente das anteriores.	64
4.13	Estatísticas para o tempo de execução dos algoritmos, em segundos.	64
4.14	Valores-p geral e das comparações pareadas entre os tempos de execução dos algoritmos, em segundos, pelo teste de Quade para determinar as melhores políticas de vacinação.	65
A.1	Conjunto de parâmetros ajustados para as heurísticas ParamILS, mGA, Revac, bNM e RPS.	104
A.2	Estatísticas para o $\log(f(x)+1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.	107
A.3	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.	107

A.4	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.	108
A.5	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.	108
A.6	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.	109
A.7	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.	109
A.8	Estatísticas para o $\log(f(x) + 1)$ dos valores de objetivo objetivos retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.	110
A.9	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.	110
A.10	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.	111
A.11	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.	111
A.12	Candidatos a conjunto de parâmetros para o algoritmo ES-BS em objetivos não ruidosos.	114
A.13	Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 231859,8154, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos não ruidosos.	115
A.14	Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-BS em objetivos não ruidosos.	115

A.15 Candidatos a conjunto de parâmetros para o algoritmo ES-QN em objetivos não ruidosos.	115
A.16 Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 8549,8055, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos não ruidosos.	115
A.17 Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-QN em objetivos não ruidosos.	116
A.18 Candidatos a conjunto de parâmetros para o algoritmo ES-AP em objetivos não ruidosos.	116
A.19 Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 21,3743, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos não ruidosos.	117
A.20 Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-AP em objetivos não ruidosos.	117
A.21 Candidatos a conjunto de parâmetros para o algoritmo ES-BS em objetivos ruidosos.	118
A.22 Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 252141,1039, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos ruidosos.	118
A.23 Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-BS em objetivos ruidosos.	119
A.24 Candidatos a conjunto de parâmetros para o algoritmo ES-QN em objetivos ruidosos.	119
A.25 Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 366736,8967, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos ruidosos.	120
A.26 Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-QN em objetivos ruidosos.	120
A.27 Candidatos a conjunto de parâmetros para o algoritmo ES-AP em objetivos ruidosos.	120
A.28 Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-45864,5972$, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos ruidosos.	120

A.29	Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-AP em objetivos ruidosos.	121
A.30	Candidatos a conjunto de parâmetros para o algoritmo ES-CC em objetivos ruidosos.	121
A.31	Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 404031,8039, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-CC em objetivos ruidosos.	122
A.32	Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-CC em objetivos ruidosos.	122
A.33	Candidatos a conjunto de parâmetros para o algoritmo ES-APCC em objetivos ruidosos.	123
A.34	Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-24167,0778$, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-APCC em objetivos ruidosos.	123
A.35	Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-APCC em objetivos ruidosos.	124
B.1	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação à cada função-objetivo no experimento sem ruído.	126
B.2	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação à cada função-objetivo no experimento sem ruído. . .	127
B.3	Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.	130
B.4	Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.	131
B.5	Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-BS nas 30 execuções.	138
B.6	Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-QN nas 30 execuções.	139
B.7	Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-AP nas 30 execuções.	140

B.8	Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-CC nas 30 execuções.	141
B.9	Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-APCC nas 30 execuções.	142

Lista de Símbolos

Algoritmos:	
ES	Estratégia Evolutiva.
ES-BS	Estratégia Evolutiva Básica (1+1)-ES.
ES-QN	Estratégia Evolutiva (1+1)-ES com busca local baseado em Quasi-Newton.
ES-AP	Estratégia Evolutiva (1+1)-ES com busca local baseado em otimização de aproximação linear-quadrática.
ES-CC	Estratégia Evolutiva (1+1)-ES com operador de comparação com confiança estatística.
ES-APCC	Estratégia Evolutiva (1+1)-ES com busca local baseado em otimização de aproximação linear-quadrática e com operador de comparação com confiança estatística.
MBI	Modelo Baseado em Indivíduo.

Parâmetros de entrada dos algoritmos:	
p_u	Fator de n que define o número de avaliações consecutivas de objetivos para atualizar o valor de desvio padrão do ES.
p_a	Fator de n que define o número máximo sucessos e falhas armazenados no arquivo em relação às soluções consecutivas do ES.
p_c	Coefficiente de atualização do valor de desvio padrão do ES na regra do 1/5.
p_l	Probabilidade de ocorrer busca local.
p_ε	Fator de diferença utilizado como critério de parada em buscas locais em relação às variáveis de decisão e valores de função-objetivo.
p_{db}	Fator do número de parâmetros de uma quadrática com hessiana simétrica cheia para determinar o número máximo de soluções armazenadas em uma base, que poderão ser usadas em uma aproximação funcional.

p_{sp}	Fator do número de parâmetros de uma quadrática com hessiana simétrica cheia para determinar o número máximo de soluções vizinhas extraídas da base para realizar uma aproximação funcional.
p_{uE}	Número máximo permitido de avaliações para uma única solução no final da execução do ES-CC e ES-APCC.
p_{α}	Nível de confiança utilizado na comparação estatística de soluções.

Símbolos gerais:

n	Dimensão do espaço de variáveis.
\mathbf{x}	Variável de decisão (vetor coluna).
\mathbf{x}'	Transposição do vetor \mathbf{x} .
\mathcal{S}	Conjunto de soluções factíveis de busca.
$\mathbf{x}^{inf}, \mathbf{x}^{sup}$	Limites inferiores e superiores que definem a região de busca \mathcal{S} .
f	Função-objetivo.
F	Função-objetivo ruidosa.
e	Número corrente de avaliações de objetivos em uma execução do ES.
E_{max}	Número máximo de avaliações de objetivos em uma execução do ES.
u	Número de avaliações consecutivas de objetivos para atualizar o valor de desvio padrão do ES.
a_{size}	Tamanho máximo da fila de sucessos e falhas das soluções geradas consecutivamente.
σ	Tamanho do raio (desvio padrão) para a geração de soluções no ES.
$\mathbf{x}_{sol}, f_{sol}$	Variáveis de decisão e avaliação da solução sol .
$U(n, \mathbf{a}, \mathbf{b})$	Gerador de um vetor aleatório no \mathbb{R}^n uniformemente distribuído entre $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.
$U(a, b)$	Gerador de um valor aleatório uniformemente distribuído entre $[a, b]$.
$N(0, 1)$	Gerador de um número Gaussiano com média 0 e desvio padrão 1.
A	Fila de sucessos e falhas das soluções geradas pelo ES.
s_l	Número de parâmetros da função linear.
s_{dh}	Número de parâmetros da quadrática com hessiana diagonal
s_{sh}	Número de parâmetros da quadrática com hessiana simétrica.
B	Base soluções avaliadas.
db_{size}	Quantidade máxima de soluções avaliadas armazenadas em B .
sp_{size}	Tamanho máximo da amostra de soluções avaliadas para se gerar uma aproximação funcional.
bl_{suc}	Booleano que indica se busca local obteve uma solução de sucesso.

ls_{qtd}	Número de execuções do Quasi-Newton na busca local com aproximação funcional, com distintos vetores iniciais.
F_{sol}	Vetor de avaliações ruidosas da solução sol .
uE	Número máximo corrente permitido de avaliações para uma única solução no ES-CC e ES-APCC.
σ_f	Desvio padrão para o experimento com funções ruidosas.
<hr/>	
Símbolos da metaotimização:	
\mathcal{A}	Uma heurística que possua um conjunto de parâmetros que define a qualidade da mesma.
$\Lambda(\boldsymbol{\pi}, f, \zeta, \kappa)$	Uma instância de \mathcal{A} com entradas $\boldsymbol{\pi}, f, \zeta$ e κ .
m	Número de metaparâmetros da heurística \mathcal{A} .
$\boldsymbol{\pi}$	Conjunto com os metaparâmetros $\pi_i, \forall \{i, \dots, m\}$ de Λ .
ζ	Semente aleatória inteira de Λ gerada por uma distribuição uniforme.
κ	Custo computacional, medido em número de avaliações de objetivo em uma execução de Λ .
Ω	Espaço amostral de todas as possíveis entradas para Λ .
Π	Conjunto de metassoluções factíveis de busca.
\mathcal{F}	Conjunto de funções-objetivo para o treinamento dos metaparâmetros.
Γ	Variável aleatória que associa a entrada $\omega \in \Omega$ em algum $\gamma = \Lambda(\omega) \in \Gamma(\Omega) \subseteq \mathbb{R}$.
$\hat{\gamma}$	Estatística de função-objetivo gerada pela amostra de melhores objetivos retornados em várias execuções de uma heurística.
MBF ou ν	<i>Mean Best Fitness</i> , o metacritério usado no processo de metaotimização.
RPS	<i>Robust Parameter Searcher</i> .
ϵ	Quantidade corrente de meta-avaliações totais nas iterações do RPS.
\mathfrak{E}	Número máximo de meta-avaliações permitidas durante todo o processo de metaotimização.
e	Quantidade corrente de meta-avaliações máxima por metassolução nas iterações do RPS.
e_{\max}	Número máximo de meta-avaliações por metassolução na última iteração do RPS.
$i_w, i_c, i_r,$ i_e, i_{oc}, i_{ic}	Índices da metasolução pior, centroide, refletida, expandida, contraída externamente e contraída internamente no simplex do RPS.
S	Vetor com as metassoluções do RPS.

RS	Random Search metastrategy.
ParamILS	Parameter Iterated Local Search.
mGA	Meta-algoritmo Genético.
Revac	Relevance Estimation and Value Calibration.
bNM	Bounded Nelder Mead.
ils_i	i -ésimo parâmetro do ParamILS, para $i \in \{1, \dots, 3\}$.
mga_i	i -ésimo parâmetro do mGA, para $i \in \{1, \dots, 4\}$.
$revac_i$	i -ésimo parâmetro do Revac, para $i \in \{1, \dots, 4\}$.
bnm_i	i -ésimo parâmetro do bNM, para $i \in \{1, \dots, 5\}$.
DE	Evolução Diferencial.
de_i	i -ésimo parâmetro da DE, para $i \in \{1, \dots, 4\}$.

Sumário

Agradecimentos	ix
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xix
Lista de Símbolos	xxv
1 Introdução	1
1.1 Contribuições da Tese	5
2 Revisão da Literatura	7
2.1 Otimização de Funções Ruidosas	7
2.2 Otimização Baseada em Aproximação Funcional	10
2.3 Estratégias Evolutivas	16
3 Metodologia	19
3.1 Problema de Otimização Ruidosa	19
3.2 Estratégia Evolutiva (1+1)-ES	20
3.3 (1+1)-ES com Busca Local Baseada em Quasi-Newton	22
3.4 Aproximação Linear e Quadrática	25
3.5 Randomized KD-Tree	26
3.6 Testes Estatísticos	28
3.6.1 Teste-t de Student	28
3.6.2 Teste de Quade	29
3.7 Algoritmos Propostos	30

3.7.1	ES-AP	30
3.7.2	ES-CC	34
3.7.3	ES-APCC	37
3.7.4	Propriedades dos Operadores Propostos	39
4	Resultados	41
4.1	Experimento 1 - Comparação dos Algoritmos em Objetivos Não Ruidosos	42
4.2	Experimento 2 - Comparação dos Algoritmos em Objetivos Ruidosos .	47
4.3	Experimento 3 - Comparação dos Algoritmos em um Problema de Controle de Epidemia	56
4.3.1	Modelos de Epidemias	58
4.3.2	Problema de Otimização	60
4.3.3	Comparação de Algoritmos	62
5	Conclusões	67
5.1	Trabalhos Futuros	72
	Referências Bibliográficas	75
	Apêndice A Sintonização dos Parâmetros de Algoritmos	93
A.1	Apresentação	93
A.2	Definição do Problema de Sintonização de Parâmetros	94
A.3	<i>Robust Parameter Searcher</i>	96
A.4	Comparações com Outras Heurísticas de Sintonização	100
A.4.1	Algoritmos Comparados	100
A.4.2	Estudo de Caso	103
A.5	Definição dos Parâmetros dos Algoritmos	112
	Apêndice B Resultados Extras	125
B.1	Apresentação	125
B.2	Comparações extras entre os Algoritmos	125
	Apêndice C <i>Multiobjective synthesis of robust vaccination policies</i>	143

Capítulo 1

Introdução

A *otimização* pode ser definida como a determinação de extremos de funções, restritas a determinados domínios. Essa definição formal, quando referida a situações de interesse prático, irá dizer respeito a funções (ditas *funções-objetivo*) que buscam representar sistemas do mundo físico, sendo que nesses casos se busca minimizar um custo, ou maximizar um ganho, que varia de acordo com uma configuração de parâmetros (as *variáveis de decisão*) que podem ser livremente escolhidos dentro de uma região factível de busca.

Assim, a determinação de soluções que tenham valor prático, no sentido de serem capazes de causar o efeito desejado de minimização do custo ou de maximização do ganho no sistema físico mediante a aplicação da configuração de parâmetros recomendada pelo método de otimização, depende em grande parte da etapa de formulação de um modelo que guarde correspondência com o sistema real a que se refere. A maior parte da literatura sobre otimização pressupõe que tal correspondência exista, sendo o modelo capaz de predizer o comportamento do sistema físico. Nos casos em que tal pressuposto seja razoável, a determinação de soluções práticas pode ser obtida pela simples determinação dos extremos das funções-objetivo.

Há alguns tipos de situações nas quais esse pressuposto deve ser relativizado. Em muitos casos, os próprios sistemas são caracterizados por uma variabilidade inerente ao seu funcionamento. Nestes casos, os modelos que os descrevem podem até ser precisos, mas os sistemas físicos incluem dinâmicas dotadas de aleatoriedade que impedem que um modelo seja capaz de realizar uma predição exata de seu comportamento. Exemplos disso são sistemas cujas entradas não são determinísticas (no sentido de que não são conhecidas a priori no momento em que se planeja a operação do sistema), tais como: sistemas de comunicação, sistemas de controle de tráfego urbano, sistemas de vendas a varejo, etc. Outros exemplos ainda se referem a sistemas cujo comportamento

é em parte governado por processos não-determinísticos, que envolvem etapas cuja resposta dinâmica é aleatorizada – e que ocorrem por exemplo em sistemas autônomos físicos (robôs exploradores) ou virtuais (oponentes virtuais em jogos eletrônicos, algoritmos evolutivos, sistemas automáticos de recomendação, etc). Em muitos casos, há uma combinação de fatores que pode envolver a imprecisão dos modelos matemáticos dos sistemas, junto com a aleatoriedade intrínseca à dinâmica envolvida. Importantes sistemas físicos se enquadram nessa categoria, como por exemplo a dinâmica de epidemias, a dinâmica de organismos biológicos e de sua resposta a fármacos, os sistemas macroeconômicos, e outros.

Nessas situações, a formulação de problemas de otimização adequados para gerarem soluções de interesse prático torna-se mais complexa. A cada realização da avaliação da função-objetivo, dado um conjunto fixo de valores para as variáveis de decisão, será obtido um valor potencialmente diferente daquele obtido em outras realizações da avaliação da função para o mesmo conjunto de variáveis. Há essencialmente duas abordagens para tratar a questão: (i) Pode-se construir uma função-objetivo substituta, montada a partir de informações analíticas sobre a estrutura das incertezas presentes, que represente uma estatística do valor da função-objetivo. Nesse caso, tal função substituta pode ser tratada como as funções isentas de incertezas que se encontram presentes nas formulações básicas da otimização. (ii) Entretanto, nem sempre encontra-se disponível a informação a respeito da distribuição estatística das variáveis aleatórias associadas ao sistema, assim como nem sempre a composição dessas variáveis com o restante do modelo conduz a distribuições analiticamente tratáveis para o valor resultante da função-objetivo. Nesses casos, resta o recurso à simulação do modelo, mesmo dotado de aleatoriedade, o que conduz a um processo de otimização baseado em uma função de mérito que é estocástica. A otimização, de fato, é feita sobre uma função-objetivo que corresponde a uma estatística dessa função de mérito (usualmente uma média), que é obtida por uma amostragem empírica sobre o modelo simulado.

Esta tese aborda precisamente este tipo de situação¹: deseja-se determinar a

¹Deve-se mencionar que há outros tipos de problemas de otimização também envolvendo incertezas sobre o sistema a ser otimizado, onde podem existir discrepâncias significativas entre os sistemas presentes no mundo real e os modelos matemático-computacionais que os representam. Os sistemas, nestes casos, podem até ter comportamento determinístico, mas os modelos que os representam, e que são utilizados no processo de otimização, não são capazes de prever tal comportamento de maneira precisa. Esse é o caso, por exemplo, da modelagem de propriedades de dispositivos que devam ser fabricados. Em virtude de imprecisões no processo de fabricação, a geometria do dispositivo físico, após fabricado, irá apresentar discrepâncias em diversos parâmetros (comprimentos, espessuras, diâmetros, etc) em relação às especificações do modelo. Também as propriedades do material (densidade, permeabilidade magnética, condutividade térmica, etc) serão um pouco diferentes das propriedades do material idealizado que se encontrava suposto na representação do modelo. Nesse caso, há um limite informacional no modelo referente à precisão (ou tolerância) dos processos envolvidos na fabricação

localização de um valor extremo de uma função-objetivo do tipo caixa-preta que corresponda à média de uma função de mérito estocástica. Os valores da função-objetivo, por sua vez, somente são acessíveis por meio de simulações, que correspondem a amostragens sobre tal função de mérito estocástica.

Um algoritmo de otimização, ao realizar uma busca sobre uma função com tais características, pode tratar as amostras obtidas da função de mérito de duas formas opostas: (i) é possível reamostrar um mesmo ponto várias vezes, na tentativa de se controlar a incerteza sobre o valor da função nesse ponto; (ii) é possível ainda amostrar apenas uma vez cada ponto, utilizando a informação assim adquirida da mesma maneira que seria utilizada caso não existisse incerteza no valor amostrado. A primeira forma essencialmente procura fazer com que as tomadas de decisão realizadas no interior do algoritmo de otimização sejam realizadas com informação equivalente à do caso isento de incertezas – ao custo de um aumento do número de avaliações de função necessário para que tal equivalência seja atingida. A segunda forma implica que as tomadas de decisão internas ao algoritmo possam ser frequentemente afetadas pela incerteza no valor da função, o que leva a uma degradação no processo de busca. Nesse segundo caso, espera-se que o algoritmo contenha mecanismos que implicitamente compensem tal degradação – o que significa que, no melhor caso, o custo da adoção de tal estratégia também pode se traduzir no aumento do número de avaliações necessário para se atingir o ótimo, em decorrência dessa degradação do mecanismo de busca. No pior caso, tal mecanismo pode levar à perda da capacidade de determinação do ponto de ótimo.

Esta tese especificamente enfoca o estudo de um mecanismo que visa reduzir o número necessário de reavaliações da função de mérito. Por meio de um mecanismo de filtragem aplicado sobre um conjunto de amostras, baseado na utilização de aproximações lineares ou quadráticas da função, constrói-se uma função substituta de caráter local capaz de guiar localmente o processo de otimização. Assim, utilizando essencialmente uma hipótese de continuidade e de suavidade da função de mérito (que se traduz na possibilidade de se realizarem aproximações lineares ou quadráticas com validade local), busca-se aqui a construção de métodos de otimização que sejam capazes de determinar o ponto de ótimo de funções de mérito estocásticas e que requeiram um número relativamente pequeno de avaliações de função.

do dispositivo. Nessa situação, podem ser definidos problemas de *otimização robusta* que visem à determinação de soluções adequadas para os diferentes sistemas possíveis sob consideração. Nesse caso, o problema de otimização pode ser formulado como uma otimização *minmax*, que corresponde à otimização do pior caso possível, ou como a otimização de algum quantil do desempenho esperado, sobre o conjunto de sistemas possíveis. Exemplos de formulações desse tipo podem ser encontrados em [Pankov et al., 2003; Bayram & Gezici, 2009; Qu et al., 2010; Marzat et al., 2013]. Esta tese não abordará esta classe de problemas.

Os estudos conduzidos no âmbito da pesquisa aqui relatada fazem uso de um tipo específico de aproximação de funções lineares e quadráticas, que é empregado para a construção de um operador de busca local para algoritmos da classe das *Estratégias Evolutivas* (ES). Os algoritmos assim constituídos são então comparados com algoritmos de Estratégias Evolutivas básicos, assim como com algoritmos de Estratégias Evolutivas dotados de outros mecanismos de busca local de outros tipos. A comparação é realizada primeiro sobre problemas com funções-objetivo determinísticas, de forma a estabelecer a capacidade de busca dos algoritmos propostos em condições-padrão. A seguir, é realizada uma comparação sobre funções-objetivo dotadas de ruídos estocásticos aditivos. Tais comparações mostram que os algoritmos propostos têm desempenho equivalente aos algoritmos com busca local não baseados nos métodos propostos, quando se utilizam as funções-objetivo isentas de ruídos. Quando são otimizadas funções-objetivo ruidosas, os testes revelam a superioridade no desempenho dos algoritmos propostos em relação aos demais. Por fim, é examinada a aplicação do operador proposto em um problema de motivação prática, relacionado com o planejamento de políticas de vacinação para o controle de epidemias. Também esse teste indica o bom desempenho desse operador.

Este texto encontra-se organizado da seguinte forma: a seção final deste capítulo apresenta uma lista detalhada das contribuições desta tese. O Capítulo 2 apresenta uma revisão bibliográfica sobre três temas que se interligam no contexto desta tese: a otimização de funções ruidosas, a otimização baseada em aproximações funcionais e as estratégias evolutivas. O Capítulo 3 apresenta a definição do problema aqui abordado e faz também a apresentação dos elementos utilizados para a construção dos operadores e algoritmos propostos. Por fim, esse capítulo também apresenta os algoritmos propostos. O Capítulo 4 apresenta os testes numéricos realizados e as análises comparativas entre os algoritmos estudados. O Capítulo 5 mostra as conclusões obtidas ao fim da pesquisa aqui relatada. O Apêndice B apresenta gráficos e tabelas extras sobre os resultados obtidos.

No que diz respeito à apresentação de resultados, o Apêndice C mostra o texto completo do artigo da referência [da Cruz et al., 2017], que apresenta o desenvolvimento de um planejamento multiobjetivo robusto para o controle de epidemias através de campanhas vacinação, que foi desenvolvido no âmbito desta tese. Nesse trabalho, foram determinados os momentos para execução de intervenções vacinais e a quantidade de vacinas a serem aplicadas em cada intervenção, de modo a minimizar o custo com a implementação do controle ao mesmo tempo em que se minimiza o dano causado pela doença na população. Nesse trabalho, o autor desta tese e seus colaboradores modelaram o problema de modo a obter: (i) um conjunto não-dominado de políticas

de controle permanente; (ii) um conjunto não-dominado de controle completo, formado pela concatenação da solução de regime permanente obtida anteriormente com uma sequência de controles específicos para a fase transitente; e (iii) uma simulação das soluções não-dominadas de controle completo, a serem analisadas pela técnica de dominância estocástica, que permite remover do conjunto aquelas políticas sensíveis à estocasticidade. Para o processo de otimização, foi utilizado um algoritmo evolutivo que utiliza uma busca local baseada na otimização de aproximações quadráticas dos objetivos originais. Esse apêndice mostra o detalhamento e a contextualização do estudo de caso que é apresentado de maneira sucinta no Experimento 3 do Capítulo 4.

Deve-se chamar ainda a atenção para uma dificuldade que teve de ser tratada no âmbito da pesquisa relacionada a este trabalho de tese. Como a questão da otimização de funções-objeto ruidosas não se trata de problema amplamente estudado, não foi possível recorrer a algoritmos-padrão com seus correspondentes valores padronizados de parâmetros para servirem de base de comparação, pois estes não existem na literatura. Foi necessário então desenvolver um mecanismo sintonizador de algoritmos que fosse capaz de produzir valores de ajustes de parâmetros obtidos nas mesmas condições, tanto para os algoritmos propostos quanto para os algoritmos que serviriam de base de comparação. A construção desse mecanismo de sintonia de parâmetros de algoritmos encontra-se relatada no Apêndice A desta tese.

1.1 Contribuições da Tese

Este trabalho apresenta as seguintes contribuições:

1. Proposição de um operador de busca local, aqui acoplado em um algoritmo (1+1)-ES, baseado na otimização de aproximações lineares ou quadráticas de funções mono-objeto. As aproximações são obtidas via regressão linear. Esse mecanismo corresponde a um desenvolvimento adicional sobre uma sequência de trabalhos do autor desta tese [da Cruz et al., 2011a,b, 2017], que aplicaram programação linear para encontrar a aproximação quadrática convexa de funções-objeto durante a busca local. Tal mecanismo de busca local se baseia em uma estrutura de dados eficiente, no caso randomized kd-tree [Muja & Lowe, 2009, 2014], para armazenar e consultar as soluções avaliadas durante as iterações do ES. Tais pesquisas são realizadas para obter a amostra que gerará a aproximação funcional. Esse mecanismo de busca local pode ser facilmente adaptado para utilização em outros tipos de algoritmos evolutivos destinados à otimização de funções de variáveis reais.

2. Elaboração de uma técnica, baseada no teste-t de Student [Montgomery, 2012], que visa comparar as avaliações múltiplas de duas soluções independentes, dado um nível de confiança. Tal procedimento é empregado apenas quando se otimizam funções ruidosas. Neste esquema, o número máximo de reavaliações por solução cresce linearmente com a quantidade de avaliações global do algoritmo evolutivo. Com base nesse artifício, é construída uma variação do operador de busca local baseado em aproximações lineares e quadráticas (ver item 1), sendo esta variação especialmente adequada para utilização no contexto da otimização de funções ruidosas.
3. Construção de um método para sintonizar parâmetros de algoritmos. Tal procedimento foi especialmente projetado para a sintonização de heurísticas destinadas à otimização de funções-objetivo. Ademais, o mesmo foi aplicado para ajustar os valores dos parâmetros das estratégias evolutivas. Esse trabalho, também realizado no contexto desta tese, é relatado no Apêndice A.

Capítulo 2

Revisão da Literatura

Neste capítulo, são apresentadas revisões bibliográficas a respeito de três temáticas que constituem aspectos centrais do presente trabalho de tese: a otimização de funções ruidosas, a otimização baseada em aproximações funcionais, e os algoritmos de estratégias evolutivas.

2.1 Otimização de Funções Ruidosas

Em computação evolutiva, problemas nos quais a função-objetivo é ruidosa são estudados desde os primórdios da área [De Jong, 1975]. Exemplos de problemas deste tipo já tratados incluem a robótica [Jakobi et al., 1995], a otimização da estrutura de redes neurais [Yao & Liu, 1997] e melhorias no layout de páginas da Internet [Penalver & Merelo, 1998].

O cenário que aparece mais comumente diz respeito a estudos em que se consideram técnicas para otimizar funções sujeitas a ruído aditivo proveniente de uma distribuição Gaussiana [Fitzpatrick & Grefenstette, 1988; Miller et al., 1995; Rekanos, 2008], que também é o caso deste trabalho de tese, ou de Cauchy [Arnold & Beyer, 2003b]. Há também situações em que se estuda e aplica ruído multiplicativo [Bioucas-Dias & Figueiredo, 2010; Zhao et al., 2014].

No trabalho de Miller & Goldberg [1996] analisou-se o efeito do ruído em distintos mecanismos de seleção em um algoritmo genético. No ambiente estudado, verificou-se ser possível prever a taxa de convergência de um algoritmo genético em domínios ruidosos caracterizados por blocos construtivos (*building blocks*) uniformemente dimensionados. Também foi verificada a possibilidade de se prever a qualidade das soluções após um certo número de gerações em ambientes ruidosos.

O artigo [Jun-hua & Ming, 2013] apresentou a construção de uma cadeia de Markov que modela um algoritmo genético elitista, sob a circunstância de que o mesmo otimiza uma função com ruído aditivo. Em tal cadeia foi provado que existe um estado absorvente, sobre o qual podem-se derivar resultados sobre os limites inferiores e superiores do número de iterações esperado para a heurística atingir uma solução ótima.

Qian et al. [2015] realizaram um estudo teórico e prático que investigou como o ruído afeta o tempo de execução de um algoritmo evolutivo simples em alguns problemas discretos do tipo *flat* e deceptivo. As técnicas de reavaliação de soluções e de seleção por limiar foram experimentadas. Neste contexto, chegou-se à conclusão de que a combinação das mesmas é mais interessante do que o uso isolado de alguma delas. Já Friedrich et al. [2015] atestaram, utilizando um algoritmo evolutivo simples em uma função com ruído aditivo Gaussiano, que o operador de recombinação é benéfico para tratar ruído, enquanto que o uso de mutação apenas leva a um desempenho consideravelmente degradado.

Encontram-se na literatura diversos trabalhos experimentais que analisaram o processo de otimização com objetivos ruidosos envolvendo algoritmos evolutivos. Em um desses trabalhos, Beyer [2000] avaliou a performance de uma versão do algoritmo genético e de uma estratégia evolutiva na função hiperesfera com ruído. Arnold & Beyer [2003a] exploraram os benefícios de se ter uma população de soluções em um ES para melhorar os resultados em ambiente ruidoso. Os autores Akimoto et al. [2015] analisaram a execução de uma versão do ES com reavaliações em problemas discretos com ruído.

É desejável que a heurística não seja enganada pelo ruído [Merelo et al., 2015] e, para isto, formas eficientes de avaliação e comparação devem ser utilizadas. Normalmente, otimiza-se um estimador da função, como por exemplo a média ou a mediana. Avaliando-se repetidamente uma solução reduz-se a variância, porém determinar o número dessas repetições é uma questão pertinente [Aizawa & Wah, 1994]. As duas categorias de reamostragem são: estática e dinâmica. Na primeira, usada em estratégias mais simples, predefine-se um número constante de reavaliações por indivíduo [Jin & Branke, 2005]. Os efeitos de tal ação são abordados em [Miller, 1997]. Na amostragem dinâmica, o número de avaliações pode variar de acordo com a precisão necessária para diferenciar, dado um critério, pares de soluções. Aspectos que possivelmente influenciam os resultados são a variabilidade dos objetivos avaliados, a periodicidade na atualização do tamanho da amostra, o número de objetivos, e a realização de análises comparativas com relação a membros da população [Rakshit et al., 2016]. Por exemplo, no artigo de Li et al. [2014] utilizou-se a estratégia de reavaliar cada solução da

população em cada geração de um algoritmo genético. No trabalho de [Merelo et al., 2016] são propostos métodos baseados em computação incremental e o uso de testes estatísticos para impor uma ordem parcial que determina o valor de aptidão das soluções da população.

No presente trabalho de tese, nas estratégias que utilizam o operador de comparação baseado em teste estatístico, o número máximo de amostragens por solução inicia em duas avaliações e cresce linearmente com o número global de avaliações de função-objetivo até uma quantidade definida previamente. Para duas soluções distintas, compara-se o par de amostras com valores de função-objetivo avaliados através do teste-t de Student [Montgomery, 2012], dado um nível de confiança. Se não houver diferença, aquela solução com maior variância de objetivo que não atingiu o limite de avaliações será reavaliada. Se o limiar desta solução já tiver sido atingido e o da outra não, então esta será reavaliada. Caso ambas já tenham sido amostradas até o número de vezes permitido, então é adotada a comparação pela média amostral.

Quando a natureza do ruído é conhecida, operadores de comparação adequados podem ser utilizados. Em heurísticas rudimentares, faz-se uma comparação de soluções usando-se apenas uma estatística estimada. Em algoritmos mais requintados, operadores mais complexos, que aproveitam melhor a informação coletada, são aplicados. Um caso particular é o trabalho de Merelo et al. [2014], que utilizou o teste de Wilcoxon em um operador de comparação de soluções no algoritmo evolutivo. Se porventura o ruído fosse gerado por uma distribuição Gaussiana, seria possível usar o teste-t para confrontar pares de soluções. Mais informações sobre diferentes formas de amostragem podem ser encontradas em [Siegmund et al., 2013] e [Rakshit et al., 2016].

Para finalizar esta subseção, a Tabela 2.1 elenca, em ordem cronológica, uma lista de aplicações ou experimentos em que algoritmos evolutivos foram utilizados na otimização de funções ruidosas.

Tabela 2.1: Trabalhos recentes com aplicações de algoritmos evolutivos na otimização de funções-objetivo ruidosas.

ES	Características Principais
Krink et al. [2004]	Levantou o desafio de melhorar os algoritmos de evolução diferencial para otimizar funções ruidosas. Apresentou um experimento em que tal heurística foi pior que outros evolutivos.
[Das et al., 2005]	Apresentou uma versão da evolução diferencial que foi significativamente melhor do que outros evolutivos no experimento apresentado.
Shir et al. [2008]	Solucionou o problema de controle quântico simulado externamente utilizando o CMA-ES.

Bayer et al. [2010]	Adotou e comparou uma versão da evolução diferencial e do CMA-ES para o gerenciamento de águas subterrâneas.
Mora et al. [2012]	Aplicou algoritmos genéticos para projetar personagens no jogo Planet Wars, cuja avaliação é dada por batalhas não determinísticas.
Krämer et al. [2014]	Comparou cinco estratégias, incluindo o CMA-ES, para otimizar os parâmetros da automatização de campos de força molecular.
[Rakshit et al., 2014]	Propôs uma versão multiobjetivo ruidosa da evolução diferencial e solucionou um problema de robótica.
[Zarifia et al., 2015]	Detectou picos neurais utilizando algoritmos genéticos em ambientes ruidosos.
Ma et al. [2015]	Mostrou uma versão de algoritmo de nuvem de partículas que possui operadores inspirados em decisão de grupo para otimizar funções com ruído.
[Eguchi et al., 2015]	Comparou versões autoadaptativas da evolução diferencial e estimou a função de distribuição de reflectância de objetos 3D.
Taghiyeh & Xu [2016]	Implementou uma versão de algoritmo de nuvem de partículas em que a partícula global é estatisticamente melhor, dado um nível de confiança.
Notsu et al. [2016]	Expôs uma estratégia que se baseia no limite superior da confiança para selecionar, no algoritmo UCT, sub-regiões mais propícias para a determinação do ótimo ruidoso.

2.2 Otimização Baseada em Aproximação Funcional

Segundo Jin & Branke [2005], as motivações para o uso de aproximações funcionais (*surrogate* ou metamodelo) em um processo evolutivo de otimização podem ser o alto custo da avaliação da função-objetivo, a indisponibilidade da forma analítica funcional, a necessidade do tratamento de ruído presente na avaliação da função, e a necessidade de se obter uma forma aproximada suavizada da geometria da função original. Jin [2011] classifica os métodos de aproximação dentro da computação evolucionária em estratégias baseadas em soluções individuais, em gerações ou em populações. Jin et al. [2000] salienta ainda que as aproximações devem ser atualizadas com novas avaliações do objetivo original, uma vez que o metamodelo pode introduzir um falso ótimo local no problema original.

Os primeiros trabalhos que utilizaram aproximações da função-objetivo apareceram durante as décadas de 1980 e 1990 [Grefenstette & Fitzpatric, 1985; Schneider et al., 1994; Yang & Flockton, 1995; Ratle, 1998; Pierret & Van den Braembussche, 1998; Bull, 1999]. Desde então, aplicações em diversas áreas se beneficiam da inserção de metamodelos no processo de evolutivo. Diversos exemplos são encontrados em projeto estrutural [Grierson & Pak, 1993; Lee & Hajela, 1996; Jin et al., 2001; Ong et al., 2003; Hüskén et al., 2005; Asouti et al., 2009], planejamento com critério de avaliação subjetiva e interativa [Biles, 1994; Johanson & Poli, 1998; Takagi, 2001; Phelps & Köksalan, 2003; Cho, 2004; Romero & Machado, 2007], e previsão de estrutura protéica [Rost & Sander, 1994; Neumaier, 1997; Cutello et al., 2006; Custódio et al., 2010].

Os métodos computacionais mais comumente usados para aproximar a função original são baseados em aproximação polinomial, em somas de funções Gaussianas e em redes neurais [Jin & Branke, 2005]. Em contraste com essa tradição, o presente trabalho de tese utiliza aproximações lineares ou quadráticas (dependendo do número de amostras disponíveis) durante uma busca local para descrever as tendências locais da função original, dada uma amostra avaliada. Tal estimativa é otimizada na região factível em um raio centralizado por uma solução selecionada. Este ótimo do problema aproximado pode ser uma boa solução para a função-objetivo original, ou pode representar uma boa indicação de direção de busca de soluções. Desse modo, pode-se melhorar a eficácia do algoritmo de busca em problemas com e sem ruído, e sem a necessidade de avaliações extras.

Há na literatura trabalhos relacionados a este. Powell [Powell, 2002, 2003, 2006] propôs métodos globais irrestritos que interpolam uma amostra avaliada de soluções por um polinômio quadrático, em cada iteração. Inicialmente, gera-se uma amostra que é avaliada e usada para a aproximação inicial. Iterativamente, o ótimo da quadrática, dentro de uma região de confiança (*trust region*), substitui uma solução da amostra. Com isso, é gerada e otimizada uma nova aproximação. Tais ações se repetem até uma condição de parada.

Wanner desenvolveu com outros autores uma busca local similar à que foi proposta aqui [Wanner et al., 2006a, 2007]. Porém, a técnica de aproximação é baseada em desigualdades matriciais lineares [Boyd & Vandenberghe, 2004] que, apesar de garantir convexidade, possui um custo computacional muito maior do que a regressão linear. A mesma autora e seus colaboradores desenvolveram, em outros trabalhos, uma busca local em otimização multiobjetivo que aproxima, pela mesma técnica, tanto a função-objetivo quanto as restrições. O problema aproximado é solucionado via *goal-attainment* [Wanner et al., 2006b, 2008a,b]. Há ainda trabalhos que utilizam dessa técnica de aproximação para tratar restrições em algoritmos evolutivos [Wanner et al.,

2005; Araujo et al., 2009; Fonseca & Wanner, 2016].

Cabe ressaltar outra diferença entre a busca local proposta neste trabalho e aquela empregada nas referências mencionadas no parágrafo anterior. Nesta tese, o armazenamento e a consulta às soluções previamente avaliadas é feita utilizando a estrutura de dados *randomized kd-tree*, o que torna a sua execução mais eficiente. Uma outra diferença é quanto à definição de vizinhança. Nos trabalhos citados, em geral define-se um raio fixo em torno de uma solução selecionada dentro do qual serão coletadas as informações de avaliação para a aproximação. Não se especifica o que ocorre se o número de amostras não atingir o mínimo necessário para o cômputo da aproximação. Na busca local proposta nesta tese, as informações de todas as soluções vizinhas mais próximas, até se atingir uma quantidade máxima, são utilizadas. Dependendo do número de vizinhos disponíveis, um tipo de aproximação será realizada (linear, quadrática com hessiana diagonal/triangular) ou não executada (caso o conjunto de amostras seja insuficiente). Além disso, como está sendo realizada uma aproximação por regressão linear, a função de aproximação pode ser não-convexa, o que pode ser útil no contexto da otimização não-linear em geral, considerado nesta tese.

O presente autor e colaboradores propuseram e compararam na publicação [da Cruz et al., 2011b] uma família de buscas locais que realizam aproximações quadráticas com hessiana semidefinida, via programação linear e via desigualdades matriciais lineares, e depois otimizam uma soma convexa de tais aproximações via programação quadrática. Estes operadores foram inseridos no algoritmo evolutivo de otimização multiobjetivo NSGA-II. As versões assim obtidas do algoritmo de otimização foram comparadas entre si e com uma versão canônica, em um conjunto de funções-objetivo. Os resultados mostraram que as versões com busca local foram superiores em relação à versão sem busca local. Também foi indicado que, embora as versões com desigualdades matriciais lineares fossem ligeiramente mais eficazes que as versões com programação linear, o custo computacional das primeiras é muito maior, sendo em geral vantajosa a adoção das últimas.

Um outro trabalho relacionado deste autor é o texto [da Cruz et al., 2011a]. Neste, uma adaptação de um algoritmo evolutivo com busca local baseada em aproximação quadrática com hessiana diagonal obtida via programação linear, conforme proposto em [da Cruz et al., 2011b], é aplicada em um problema multiobjetivo de controle de epidemias através de campanhas de vacinação. Dado um horizonte de tempo, definem-se as taxas de vacinação e os intervalos de tempo entre campanhas de vacinação, de modo a minimizar concomitantemente o custo de implementação e o prejuízo causado pela doença. O modelo da propagação da epidemia é baseado em um sistema de equações diferenciais. Também é comparada a eficácia do algoritmo proposto em relação

à versão sem busca local. Os resultados indicaram que as soluções geradas da versão com busca local são mais diversificadas e dominam a maioria das soluções geradas pelo método canônico.

O artigo [da Cruz et al., 2017], também do autor, é uma evolução do trabalho discutido no parágrafo anterior, já tendo sido desenvolvido no âmbito da pesquisa realizada para a elaboração desta tese. O problema multiobjetivo de controle de epidemias por vacinação é solucionado em três etapas. Na primeira, determina-se um conjunto não dominado de soluções para o controle do sistema em regime permanente, as quais devem ser aplicadas após o término do regime transitório do sistema dinâmico que ocorre no início da epidemia. Na segunda parte, determinam-se as políticas completas de controle, incluindo o controle para o regime transitório. Para isto, seleciona-se uma solução não dominada da primeira etapa para ser concatenada ao final das soluções que representam políticas para o regime transitório. Com isto, atinge-se um conjunto de estratégias não dominadas para vacinação que visa: (i) em um primeiro momento reduzir os infectados de forma rápida, com um conjunto de ações de controle descrito por taxas de vacinação e intervalos de tempo entre campanhas não necessariamente constantes; e (ii) após a política transiente, controlar o sistema dinâmico visando à redução plena da população de infectados ao longo do tempo com uma taxa de vacinação e intervalo de tempo entre campanhas constantes. Na terceira parte, o conjunto não dominado final é simulado em um Modelo Baseado em Indivíduo (MBI) para observar o comportamento das soluções quando sujeitas à aleatoriedade que é intrínseca ao sistema físico, possibilitando remover do conjunto de soluções aquelas políticas estocasticamente dominadas. Desse conjunto robustamente não dominado, foram extraídas informações sobre as probabilidades de erradicação e de se ter o número de infectados abaixo de determinado limiar. Para a obtenção das políticas de controle foi utilizada uma versão do NSGA-II com busca local baseada na otimização da soma convexa de quadráticas semidefinidas que aproximam as funções-objetivo via programação linear. Nesta estratégia também foi usada uma estrutura de dados em árvore para armazenar soluções previamente avaliadas para serem consultadas via vizinhos mais próximos, quando a aproximação fosse executada. Uma comparação entre o NSGA-II dotado dos operadores propostos e a versão básica desse algoritmo em diferentes cenários de epidemias foi realizada, indicando que a heurística proposta sempre obtém um conjunto não dominado de melhor qualidade, usando um mesmo orçamento para a avaliação de objetivos.

Outros textos que abordam a construção explícita de funções quadráticas como uma ferramenta em diversas aplicações são apresentados a seguir, em ordem aproximadamente cronológica.

Algumas referências relativamente antigas encontradas sobre a construção de funções quadráticas aproximantes como mecanismo auxiliar em procedimentos de otimização são os textos de Shafer [1974], Smith & Schmidt [1977] e Minkin [1983]. O primeiro apresentou um método baseado em derivação de segunda ordem e mostrou aplicações analíticas. O segundo aplicou sucessivas aproximações para otimizar um sistema simulado computacionalmente. O terceiro propôs uma técnica para encontrar um limite para o erro na aproximação quadrática da função de log-verossimilhança.

Kiwiel [1984] e Werner [1986] utilizaram um método de aproximação quadrática para encontrar a solução ótima de modelos não lineares. Bhattacharyya & Willment [1988] trataram com quadráticas o problema do deslocamento de descontinuidade e apresentaram diversas vantagens do uso da mesma. Biernacki et al. [1989] e Bandler et al. [1991] utilizaram aproximação quadrática para modelar a resposta de circuitos, resultando uma interpolação mais simples e eficaz. Mulvey et al. [1992] criaram um método com aproximação quadrática diagonal que trata subproblemas de programação linear com alta dimensão.

Cao & Voth [1995] e Rosenfelder & Schreiber [2001] aplicaram a aproximação quadrática para modelar sistemas físicos. O primeiro texto descreve uma função de energia potencial sobre um oscilador harmônico. O segundo trata a ação retardada dos polarons ao solucionar equações não lineares variacionais com um método iterativo. Jamai & Damil [2001] utilizaram aproximação quadrática para aumentar o intervalo válido de séries vetoriais.

Den Hertog et al. [2002] criou um método de programação semidefinida que determina aproximações quadráticas convexas reais. Rosen & Marcia [2004] apresentaram um modelo de programação linear para obter aproximações quadráticas convexas. Marcia et al. [2005] aplicaram esta técnica em um processo iterativo para encontrar a solução de um problema de acoplamento de proteínas. Wang [2007] generalizou o escore de Fisher e os métodos de Gauss-Newton em um modo único, aplicando aproximação quadrática iterativamente para a computação da máxima verossimilhança.

Para finalizar esta subseção, a Tabela 2.2 cataloga, por ano, uma seleção de outras aplicações recentes que utilizaram aproximações funcionais explícitas em procedimentos de otimização.

Tabela 2.2: Trabalhos recentes com aplicações de otimização com modelos de aproximação funcional presentes na literatura.

ES	Características Principais
----	----------------------------

Logvinenko [2003]	Usou aproximação quadrática para modelar a identificação de analisadores e canais para representação da visão humana.
Nasayama et al. [2006]	Utilizou aproximação quadrática na construção de um método sintetizador para a geração de funções numéricas programáveis.
Ong et al. [2006]	Aplicou um algoritmo genético que aproxima o objetivo original via função de base radial para um projeto robusto em aerodinâmica.
Xue et al. [2006]	Estimou via aproximação quadrática a região de estabilidade sistemas de energia de múltiplas máquinas com modelo de rede reduzido.
Boubekeur & Schlick [2007]	Desenvolveu, em computação gráfica, um aproximador quadrático para gerar a subdivisão de superfícies.
Samad et al. [2008]	Utilizou três modelos de aproximação funcional na otimização da forma de uma lâmina do rotor 37 da Nasa.
Liao et al. [2008]	Otimizou de forma multiobjetivo, com NSGA-II e um modelo de regressão, um projeto para segurança de automóveis.
Gao & Wang [2009]	Elaborou uma heurística com simulações inspirada em design de experimento para reduzir a distorção em modelos de capa para celulares.
Groenwold & Etman [2010]	Usou aproximação quadrática na otimização de topologias estruturais.
Li et al. [2010]	Empregou função de base radial para aproximar o objetivo original na otimização de um perfil para embalar e moldar via injeção.
Kimura [2010]	Otimizou um modelo contínuo no tempo, com aproximação quadrática, que determina valores nas concessões de ações executivas.
Wang et al. [2010]	Executou um processo otimização, com regressão linear, que soluciona um problema de aerodinâmica para exaustão de turbina.
Benigno & Woodford [2012]	Solucionou um problema dinâmico não linear de política sequencial ótima utilizando aproximação linear-quadrática.
Müller et al. [2013]	Implementou um algoritmo que utiliza um modelo de aproximação com função de base radial para otimização não linear inteira mista.
Mendes et al. [2013]	Utilizou programação genética multiobjetivo robusta com modelos aproximativos para otimizar um problema de magnetostática.
Hornng & Lin [2013]	Empregou uma rede neural para guiar um algoritmo evolutivo ao solucionar o problema de simulação estocástica combinatória em tempo real.
Xie et al. [2013]	Aplicou aproximação quadrática para auxiliar no problema de otimização de alocação em tráfego.
Chen et al. [2014]	Adotou uma família de técnicas de aproximação na otimização de funções custosas para a taxa ótima de pedágio em redes de transporte.

Hussain et al. [2015]	Usou aproximação polinomial em um algoritmo genético multiobjetivo para projetar um sistema aquífero sujeito a intrusão de água marinha.
Gutiérrez-Antonio & Briones-Ramírez [2016]	Empregou redes neurais para guiar a otimização multiobjetivo em um processo de destilação para uma aplicação em química.
Koziel & Bekasiewicz [2016]	Aplicou um modelo de aproximação local quadrática sem termos cruzados em um projeto de antenas multiobjetivo.

2.3 Estratégias Evolutivas

A Estratégia Evolutiva, ES, heurística inspirada na evolução biológica, foi estudada primeiramente na Alemanha durante década de 1960 [Rechenberg, 1965; Schwefel, 1965]. Em geral, tal algoritmo possui operadores de variação (mutação Gaussiana), com parâmetros autoadaptativos, e seleção, que atuam sobre um conjunto de soluções pai e filho para otimizar funções caixa-preta [Beyer & Schwefel, 2002].

Via de regra para as versões clássicas, adota-se a notação $(\mu/\rho^+\lambda)$ -ES, sendo μ o número de pais e $\rho \leq \mu$ a quantidade de pais selecionada para se gerar λ filhos. O sinal “+” indica junção das populações para se escolher os próximos pais, e “,” para apenas usar filhos na seleção. Quando $\mu = \rho$, é comum omitir ρ [Beyer & Schwefel, 2002].

Este trabalho adotará o algoritmo $(1+1)$ -ES como base para estudos sobre os efeitos de operadores construídos a partir de aproximações quadráticas, visando o tratamento de funções ruidosas. A escolha de tal algoritmo-base é devida à simplicidade e ao padrão de funcionamento bem definido desse algoritmo, o que facilita a observação do efeito produzido pelos operadores de busca local propostos neste trabalho. Deve-se notar que, mesmo com raio de busca σ constante (sem a regra do $1/5$), há provas sobre a convergência global desta estratégia em certas circunstâncias [Rechenberg, 1973]. Não se estabelece aqui o objetivo da estruturação de um método heurístico de alto desempenho. Se fosse este o caso, a busca local aqui proposta poderia ser acoplada a estratégias mais complexas como o NBIPOPcMA [Loshchilov, 2013], L-SHADE [Tanabe & Fukunaga, 2014], SPS-L-SHADE-EIG [Guo et al., 2015], UMOEAI [Elsayed et al., 2016], que são métodos vencedores de diferentes competições de otimização do IEEE Congress on Evolutionary Computation dos anos de 2013, 2014, 2015 e 2016. Entretanto, para o propósito aqui definido de análise das propriedades dos operado-

res propostos no contexto de funções ruidosas, é mais adequada a utilização de um algoritmo mais simples, com propriedades mais bem conhecidas.

A Tabela 2.3 lista, em ordem cronológica, uma compilação de diversas versões do ES propostas para a otimização de funções mono-objetivo.

Tabela 2.3: Versões comuns do ES encontradas na literatura.

ES	Características Principais
(μ, λ) -ES	Primeiro ES que adapta uma matriz de covariância [Schwefel, 1981].
DR1	Reduziu a aleatoriedade da (μ, λ) -ES ao usar a largura da melhor mutação para controlar o passo adaptativo e pela adoção de um parâmetro que amortece flutuações indesejadas [Ostermeier et al., 1994a].
DR2	Na mutação, incluiu parâmetros para o tamanho do passo global e local, e considerou o comprimento do passo adaptativo das melhores soluções de iterações prévias para a geração dos filhos [Ostermeier et al., 1994b].
$(1, \lambda)$ -GSA-ES	Executou a mutação com uma distribuição normal, conforme adaptação de uma matriz de covariância arbitrária [Hansen et al., 1995].
CMA-ES	Implementou a adaptação da matriz de covariância e a adaptação do tamanho do passo cumulativo que permitiram o crescimento da probabilidade da geração do mesmo passo de mutação, prover um controle direto no mecanismo para a taxa de alterações nos parâmetros, e, no caso de seleção aleatória, manter inalterados os valores dos parâmetros da estratégia [Hansen & Ostermeier, 1996, 2001].
LS-CMA-ES	Uma implementação do CMA cuja matriz de covariância é adaptada via estimação de mínimos quadrados [Auger et al., 2004].
LR-CMA-ES	Extensão do CMA-ES com reinicialização local, quando uma estagnação é identificada [Auger & Hansen, 2005a].
IPOP-CMA-ES	Extensão do LR-CMA-ES com incremento no tamanho da população antes da reinicialização [Auger & Hansen, 2005b].
$(1+1)$ -Cholesky-CMA-ES	Versão do CMA-ES que adapta a matriz de covariância através da decomposição de Cholesky [Igel et al., 2006].
Active-CMA-ES	Utilizou também os filhos para adaptar a matriz de covariância, inclusive usando os piores com peso negativo [Jastrebski & Arnold, 2006].
sep-CMA-ES	Reduziu a complexidade do CMA-ES em $O(n)$ usando uma matriz de covariância diagonal, que não é capaz de gerar mutações correlacionadas [Ros & Hansen, 2008].
xNES	Adaptou uma $(1, \lambda)$ -ES para que os parâmetros das estratégias fossem atualizados com o gradiente natural [Glasmachers et al., 2010].

iCMAES-ILS	Realizou uma combinação entre o IPOP-CMA-ES e a heurística ILS [Liao & Stuetzle, 2013].
NBIPOP-aCMA-ES	É uma versão que duplica a população no processo de reinicialização e diminui o tamanho do passo por um fator dado [Loshchilov, 2013].

Capítulo 3

Metodologia

Neste capítulo são inicialmente apresentadas as definições formais relacionadas com o problema a ser tratado nesta tese. A seguir, é apresentado o algoritmo-base do tipo Estratégia Evolutiva a partir do qual serão construídos os operadores propostos neste trabalho. Uma versão desse algoritmo, empregando busca local, a ser empregada como base de comparação com os algoritmos propostos nesta tese, é também discutida. É apresentada depois, a técnica de aproximação linear ou quadrática de funções que será empregada para a construção dos operadores de busca local aqui propostos. A seguir, é descrita a estrutura de dados denominada *randomized kd-tree*, usada em duas heurísticas propostas neste trabalho. São então explicados os dois testes estatísticos usados nesta tese.

Este capítulo, por fim, apresenta as contribuições principais desta tese, que são sintetizadas em três variações de estratégias evolutivas produzidas neste trabalho. A primeira, é construída com a busca local baseada na otimização de aproximação linear-quadrática e que utiliza a *randomized kd-tree* para armazenar e consultar soluções avaliadas durante a evolução. A segunda, utiliza um operador de comparação baseado no teste-t de Student [Montgomery, 2012] com reamostragem dinâmica. A terceira, combina a utilização do método de busca local baseado em aproximação linear-quadrática com o operador de reamostragem e comparação que se baseia no teste-t.

3.1 Problema de Otimização Ruidosa

Um problema de otimização com um único objetivo pode ser representado através da Equação 3.1. Neste problema, busca-se o vetor coluna de variáveis de decisão $\mathbf{x} = [x_1, \dots, x_n]' \in \mathbb{R}^n$ que pertence ao conjunto de soluções factíveis \mathcal{S} definido por um conjunto de restrições de desigualdade, $g_i(\mathbf{x}) \leq 0, \forall i \in \{1, \dots, G\}$, igualdade,

$h_i(\mathbf{x}) = 0, \forall i \in \{1, \dots, H\}$, ou do tipo *caixa*, $x_i^{inf} \leq x_i \leq x_i^{sup}, \forall i \in \{1, \dots, n\}$, $\mathbf{x}^{inf}, \mathbf{x}^{sup} \in \mathbb{R}^n$. Toda solução pertencente a \mathcal{S} é dita ser factível, caso contrário, infactível. Nos experimentos computacionais realizados neste trabalho são consideradas apenas restrições do tipo caixa. O critério de qualidade das soluções é definido pela função-objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$, considerada aqui como uma função caixa-preta, para a qual se supõe o acesso apenas por meio de avaliações de seu valor em pontos isolados do espaço \mathbb{R}^n . Deseja-se encontrar a melhor solução factível $\mathbf{x}^* \in \mathcal{S}$, chamada de ótimo, com o menor valor possível de $f, f(\mathbf{x}^*)$.

$$\begin{aligned} \mathbf{x}^* &= \arg \min f(\mathbf{x}) \\ \text{sujeito a: } \mathbf{x} &\in \mathcal{S} \subseteq \mathbb{R}^n \end{aligned} \tag{3.1}$$

Quando a avaliação da função-objetivo é somada a um ruído aleatório, $F(\mathbf{x}, \xi) = f(x) + \xi$, sendo $\xi \sim \Xi$ uma variável aleatória proveniente de uma distribuição qualquer, o problema de otimização representado pela Equação 3.2 consiste em encontrar a solução que otimize um estimador \mathcal{E} :

$$\begin{aligned} \mathbf{x}^* &= \arg \min \mathcal{E}[F(\mathbf{x}, \xi)] \\ \text{sujeito a: } \mathbf{x} &\in \mathcal{S} \subseteq \mathbb{R}^n \end{aligned} \tag{3.2}$$

Esse estimador, que representa uma função de mérito, pode ser definido como um quantil ou uma média. Tal problema de otimização é designado aqui por *problema de otimização ruidosa*.

Para maiores detalhes sobre teoria da otimização, são sugeridos os textos de Boyd & Vandenberghe [2004], Takahashi [2007] e Bertsekas [2016]. Para maiores detalhes sobre otimização em ambientes ruidosos, são indicados os trabalhos de Jin & Branke [2005], Merelo et al. [2015] e Rakshit et al. [2016].

3.2 Estratégia Evolutiva (1+1)-ES

A heurística Estratégia Evolutiva (1+1)-ES, representada na Figura 3.1 e designada aqui por ES-BS (*Evolution Strategy - BaSic version*), é o algoritmo utilizado neste trabalho tanto como bloco construtor para as heurísticas aqui propostas como também para servir como base de comparação. A mesma foi selecionada devido à sua simplicidade e às propriedades bem definidas de funcionamento que possui, conforme discutido por Beyer & Schwefel [2002]. Essa versão canônica detém três parâmetros de entrada: $p_u \in [0,8; 1,2]$ é o fator de n que define o número u de avaliações consecutivas de objetivos para atualizar o raio σ (desvio padrão) para geração da solução filho no

```

Entrada:  $p_u, p_a, p_c$ 
Dados:  $n, \mathbf{x}^{inf}, \mathbf{x}^{sup}, E_{max}, f$ 
1 início
2    $u \leftarrow \lfloor p_u \cdot n \rfloor;$ 
3    $a_{size} \leftarrow \lfloor p_a \cdot n \rfloor;$ 
4    $\sigma \leftarrow 1;$ 
5    $\mathbf{x}_P \leftarrow U(n, \mathbf{x}^{inf}, \mathbf{x}^{sup});$ 
6    $f_P \leftarrow f(\mathbf{x}_P); E \leftarrow 1;$ 
7    $A \leftarrow \emptyset;$ 
8   enquanto  $E < E_{max}$  faça
9      $\mathbf{x}_C \leftarrow \mathbf{x}_P + \sigma \cdot [N_1(0, 1), \dots, N_n(0, 1)]';$ 
10     $f_C \leftarrow f(\mathbf{x}_C); E \leftarrow E + 1;$ 
11    se  $f_C < f_P$  então
12       $\mathbf{x}_P \leftarrow \mathbf{x}_C; f_P \leftarrow f_C;$ 
13       $A.Insere(\text{sucesso});$ 
14    senão
15       $A.Insere(\text{falha});$ 
16    fim
17    se  $A.Tamanho() > a_{size}$  então  $A.RemovePrimeiro();$ 
18    se  $E \bmod u = 0$  então
19       $r_t \leftarrow A.TotalDe(\text{sucesso})/A.Tamanho();$ 
20       $\sigma \leftarrow \left\{ \begin{array}{ll} \sigma \cdot p_c & \text{se } r_t < 1/5 \\ \sigma/p_c & \text{se } r_t > 1/5 \\ \sigma & \text{se } r_t = 1/5 \end{array} \right\};$ 
21    fim
22  fim
23 fim
24 retorna  $\mathbf{x}_P$ 

```

Figura 3.1: Estratégia Evolutiva (1+1)-ES designada por ES-BS.

ES; $p_a \in [5; 15]$ é o fator de n que determina o máximo, a_{size} , de sucessos e falhas consecutivas das soluções armazenadas na fila A ; e $p_c \in [0,5; 1]$ é a constante usada na regra do 1/5 para atualizar o valor de σ .

Este algoritmo inicialmente gera uniformemente uma solução pai, $\mathbf{x}_P \in \mathcal{S}$, que é avaliada pelo objetivo f . Em seguida, são inicializadas a variável E , responsável pelo número de avaliações, e a fila de sucessos e falhas das soluções sucessivas, A , cujo tamanho máximo é a_{size} . No laço de repetição, que ocorre enquanto E não atinge E_{max} , é gerada uma solução filho somando-se um vetor Gaussiano de média $\mathbf{0}$ e desvio padrão σ com \mathbf{x}_P . Tal filho é avaliado e comparado com a solução pai. Se for melhor, então este será o novo pai e um “sucesso” é inserido na fila A . Caso contrário, uma “falha” é inserida. Se porventura A atingir o máximo de entradas, então remove-se

a primeira entrada. No final do laço de repetição testa-se se chegou o momento de atualizar o desvio padrão σ . Em caso positivo, este é corrigido de acordo com a regra do 1/5, conforme a linha 20 do algoritmo apresentado na Figura 3.1.

Para maiores informações sobre inspirações biológicas, evolução histórica, versões mais avançadas, outras propriedades técnicas e teoremas acerca do ES indicam-se os textos de Beyer & Schwefel [2002], Bäck et al. [2013] e Hansen et al. [2015].

3.3 (1+1)-ES com Busca Local Baseada em Quasi-Newton

A Figura 3.2 retrata o algoritmo ES-QN (*Evolution Strategy with local search based on Quasi-Newton algorithm*), que se trata de uma modificação do algoritmo ES-BS com a inclusão de uma busca local baseada em um algoritmo Quasi-Newton (ver linhas 18 a 20). Tal busca local é adequada para a otimização de funções não lineares para as quais o gradiente e a hessiana não se encontram disponíveis *a priori*. Além dos parâmetros presentes no algoritmo ES-BS, há dois outros parâmetros adicionais: $p_l \in [0,01; 0,5]$ representa a probabilidade de ocorrer a busca local; e $p_\epsilon \in [1e-4; 1]$ é usado no critério de parada como um valor de tolerância para findar a busca local e também no cálculo do vetor gradiente via diferenças finitas.

Após a análise do tamanho da fila de sucessos/insucessos, sorteia-se um número proveniente da distribuição uniforme entre 0 e 1. Se tal valor aleatório for menor que p_l , a busca local ocorrerá através do algoritmo Quasi-Newton, delineado na Figura 3.3. Este, respectivamente, receberá como parâmetros a solução pai e a respectiva avaliação de objetivo, a função-objetivo, os limites da região de busca, o valor de erro tolerável, e os números de avaliações corrente e máximo permitidos.

A estimação iterativa da hessiana é realizada utilizando o método BFGS. Primeiramente, ela é iniciada como uma matriz identidade na linha 4, e depois é seguidamente atualizada conforme a linha 16. A correção das entradas desta matriz usa a hessiana corrente, \mathbf{H}_k , e os vetores de diferença entre soluções consecutivas, \mathbf{v}_k , e entre gradientes estimados, \mathbf{r}_k . Note que \mathbf{r}'_k e \mathbf{v}'_k são as transposições de \mathbf{r}_k e \mathbf{v}_k , respectivamente.

Ressalta-se aqui a obtenção do vetor gradiente aproximado, \mathbf{g}_k , nas linhas 6 e 13, usado para determinar a direção de busca \mathbf{d}_k , juntamente com a hessiana. Tal gradiente é obtido via técnica das diferenças finitas do tipo “à frente” (*forward*), que implica no cômputo de n avaliações adicionais de função-objetivo.

Para sair do laço de repetição da busca local, deve-se atender pelo menos uma de três condições em *CriterioDeParada*: analisa-se o número de avaliações de função-

```

Entrada:  $p_u, p_a, p_c, p_l, p_\varepsilon$ 
Dados:  $n, \mathbf{x}^{inf}, \mathbf{x}^{sup}, E_{max}, f$ 
1 início
2    $u \leftarrow \lfloor p_u \cdot n \rfloor$ ;
3    $a_{size} \leftarrow \lfloor p_a \cdot n \rfloor$ ;
4    $\sigma \leftarrow 1$ ;
5    $\mathbf{x}_P \leftarrow U(n, \mathbf{x}^{inf}, \mathbf{x}^{sup})$ ;
6    $f_P \leftarrow f(\mathbf{x}_P)$ ;  $E \leftarrow 1$ ;
7    $A \leftarrow \emptyset$ ;
8   enquanto  $E < E_{max}$  faça
9      $\mathbf{x}_C \leftarrow \mathbf{x}_P + \sigma \cdot [N_1(0, 1), \dots, N_n(0, 1)]'$ ;
10     $f_C \leftarrow f(\mathbf{x}_C)$ ;  $E \leftarrow E + 1$ ;
11    se  $f_C < f_P$  então
12       $\mathbf{x}_P \leftarrow \mathbf{x}_C$ ;  $f_P \leftarrow f_C$ ;
13       $A.Insere(\text{sucesso})$ ;
14    senão
15       $A.Insere(\text{falha})$ ;
16    fim
17    se  $A.Tamanho() > a_{size}$  então  $A.RemovePrimeiro()$ ;
18    se  $U(0, 1) < p_l$  então
19       $[\mathbf{x}_L, f_L, E] \leftarrow QuasiNewton(\mathbf{x}_P, f_{\mathbf{x}_P}, f, \mathbf{x}^{inf}, \mathbf{x}^{sup}, p_\varepsilon, E, E_{max})$ ;
20      se  $f_L < f_P$  então  $\mathbf{x}_P \leftarrow \mathbf{x}_L$ ;  $f_P \leftarrow f_L$ ;
21    fim
22    se  $E \bmod u = 0$  então
23       $r_t \leftarrow A.TotalDe(\text{sucesso})/A.Tamanho()$ ;
24       $\sigma \leftarrow \left\{ \begin{array}{ll} \sigma \cdot p_c & \text{se } r_t < 1/5 \\ \sigma/p_c & \text{se } r_t > 1/5 \\ \sigma & \text{se } r_t = 1/5 \end{array} \right\}$ ;
25    fim
26  fim
27 fim
28 retorna  $\mathbf{x}_P$ 

```

Figura 3.2: (1+1)-ES com busca local baseada em Quasi-Newton designada por ES-QN.

```

Entrada:  $\mathbf{x}, f_{\mathbf{x}}, f, \mathbf{x}^{inf}, \mathbf{x}^{sup}, \varepsilon, E, E_{max}$ 
Dados:  $n$ 
1 início
2    $k \leftarrow 0;$ 
3    $\mathbf{x}_k \leftarrow \mathbf{x};$ 
4    $f_{\mathbf{x}_k} \leftarrow f_{\mathbf{x}};$ 
5    $\mathbf{H}_k \leftarrow \mathbf{I};$ 
6    $[\mathbf{g}_k, E] \leftarrow \text{Gradiente}(\mathbf{x}_k, f_{\mathbf{x}_k}, f, \varepsilon, E);$ 
7   enquanto (verdadeiro) faça
8      $\mathbf{d}_k \leftarrow -\mathbf{H}_k \mathbf{g}_k;$ 
9      $\delta_k \leftarrow \arg \min_{\delta} f(\mathbf{x}_k + \delta \mathbf{d}_k)$  s. a:  $(\mathbf{x}^{inf} \leq \mathbf{x}_k + \delta \mathbf{d}_k \leq \mathbf{x}^{sup});$ 
10     $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \delta_k \mathbf{d}_k;$ 
11     $f_{\mathbf{x}_{k+1}} \leftarrow f(\mathbf{x}_{k+1});$ 
12     $E \leftarrow E + 1;$ 
13     $[\mathbf{g}_{k+1}, E] \leftarrow \text{Gradiente}(\mathbf{x}_{k+1}, f_{\mathbf{x}_{k+1}}, f, \varepsilon, E);$ 
14     $\mathbf{v}_k \leftarrow \mathbf{x}_k - \mathbf{x}_{k+1};$ 
15     $\mathbf{r}_k \leftarrow \mathbf{g}_k - \mathbf{g}_{k+1};$ 
16    se  $\text{CritérioDeParada}(E, E_{max}, \mathbf{v}_k, \mathbf{r}_k, \varepsilon)$  então
17      saia do enquanto;
18    fim
19     $\mathbf{H}_{k+1} \leftarrow \mathbf{H}_k + \left(1 + \frac{\mathbf{r}'_k \mathbf{H}_k \mathbf{r}_k}{\mathbf{r}'_k \mathbf{v}_k}\right) - \frac{\mathbf{v}_k \mathbf{r}'_k \mathbf{H}_k + \mathbf{H}_k \mathbf{r}_k \mathbf{v}'_k}{\mathbf{r}'_k \mathbf{v}_k};$ 
20     $k \leftarrow k + 1;$ 
21  fim
22 fim
23 retorna  $[\mathbf{x}_k, f_{\mathbf{x}_k}, E]$ 

```

Figura 3.3: Função *QuasiNewton* utilizada para encontrar um mínimo local de uma função não linear.

objetivo E (também considerando as avaliações no interior do cálculo do gradiente e seção áurea) que deve ser maior ou igual ao limite especificado E_{max} , e uma das normas Euclidianas de diferenças entre duas respectivas soluções ou dois respectivos gradientes, ser menor ou igual ao limiar ε .

Menciona-se que não foi necessário o uso de métodos de penalidade uma vez que, antes da execução do algoritmo da seção áurea, projeta-se o ponto extremo, contrário à direção \mathbf{d}_k , no limite da região de busca. Para maiores detalhes sobre os métodos Quasi-Newton, recomenda-se a leitura de Takahashi [2007].

3.4 Aproximação Linear e Quadrática

Dado um conjunto de soluções avaliadas, os algoritmos propostos nesta tese aproximam uma função-objetivo por uma função linear ou quadrática no procedimento de busca local. A partir dessa aproximação é possível determinar o ponto de ótimo da função de aproximação que, por construção, tende a capturar localmente a tendência da função original. A solução encontrada pela busca local é retornada para o algoritmo evolutivo. Dessa forma, espera-se que tal solução seja razoável para o problema original, de modo a melhorar a convergência da heurística otimizadora.

Seja a função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $B = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \dots, (\mathbf{x}_K, f(\mathbf{x}_K))\}$ um conjunto de pares ordenados, dado por vetores com suas respectivas avaliações (B aqui também é chamado de base ou histórico de soluções avaliadas). A aproximação quadrática de f , $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$, conforme Equação 3.3, sendo $c_0 \in \mathbb{R}$ o termo independente, $\mathbf{c} \in \mathbb{R}^n$ o vetor coluna que representa o termo linear e $\mathbf{C} \in \mathbb{R}^{n \times n}$ a matriz hessiana, é tal que o conjunto de parâmetros $\mathbf{p}_{\tilde{f}} = \begin{bmatrix} c_0 & \dots & c_{(n^2-n)/2} \end{bmatrix}$ minimize uma norma para o vetor de erro $\mathbf{e} = \begin{bmatrix} \tilde{f}(\mathbf{x}_1) - f(\mathbf{x}_1) & \dots & \tilde{f}(\mathbf{x}_K) - f(\mathbf{x}_K) \end{bmatrix}$.

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= c_0 + \mathbf{c}'\mathbf{x} + 0,5\mathbf{x}'\mathbf{C}\mathbf{x} \\ &= c_0 + \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} \mathbf{x} + 0,5\mathbf{x}' \begin{bmatrix} c_{n+1} & c_{2n+1} & \dots & c_{3n-2} & c_{3n-1} \\ c_{2n+1} & c_{n+2} & \dots & c_{4n-4} & c_{4n-3} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{3n-2} & c_{4n-4} & \dots & c_{2n-1} & c_{(n^2+3n)/2} \\ c_{3n-1} & c_{4n-3} & \dots & c_{(n^2+3n)/2} & c_{2n} \end{bmatrix} \mathbf{x} \end{aligned} \quad (3.3)$$

Neste trabalho, a aproximação é realizada por regressão linear [Montgomery, 2012] de acordo com as seguintes regras em relação ao número de soluções avaliadas durante a execução do algoritmo: se houver $K > (n^2 + 3n)/2 + 1$ amostras, uma aproximação quadrática considerando uma matriz hessiana simétrica arbitrária será calculada. Se $2n+1 < K \leq (n^2 + 3n)/2 + 1$, será calculada uma aproximação quadrática com hessiana diagonal (removem-se os parâmetros de fora da diagonal de \mathbf{C}). Finalmente, se $n+1 < K \leq 2n+1$, uma aproximação linear será determinada (remove-se a parte quadrática do modelo). Usando a Equação 3.3, realiza-se a aproximação via regressão linear utilizando a Equação 3.4, que é solucionada pelo método de Householder com

decomposição QR e pivotação completa [Björck, 1996].

$$\mathbf{y} = \mathbf{X}\mathbf{p} + \boldsymbol{\varepsilon} \Rightarrow \hat{\mathbf{p}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y}, \quad \text{em que:}$$

$$\mathbf{y} = \begin{bmatrix} f(\mathbf{X}_1) \\ \vdots \\ f(\mathbf{X}_K) \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon^{(1)} \\ \vdots \\ \varepsilon^{(K)} \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} c_0 \\ \vdots \\ c_{(n^2+3n)/2} \end{bmatrix} \quad (3.4)$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} & x_1^{(1)2} & \dots & x_n^{(1)2} & x_1^{(1)}x_2^{(1)} & \dots & x_{n-1}^{(1)}x_n^{(1)} \\ & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_1^{(K)} & \dots & x_n^{(K)} & x_1^{(K)2} & \dots & x_n^{(K)2} & x_1^{(K)}x_2^{(K)} & \dots & x_{n-1}^{(K)}x_n^{(K)} \end{bmatrix}$$

O modelo de regressão linear foi escolhido devido à vantagem do baixo custo computacional quando comparado aos modelos de programação linear [da Cruz et al., 2011b] ou quadrática e desigualdades matriciais lineares [Fonseca & Wanner, 2016]. Por outro lado, estes três últimos são capazes de garantir a convexidade da quadrática, algo inexistente neste modelo de regressão. Deste modo, como a hessiana da função quadrática pode ser não semidefinida positiva, o problema de encontrar o ótimo global é NP-Difícil [Van Thoai, 2005], pois existe a possibilidade de existirem mínimos locais quando há restrições limitando as variáveis. Inclusive, há situações em que a função original é convexa e a regressão linear pode retornar uma aproximação quadrática não convexa [Den Hertog et al., 2002]. Desse modo, a busca local aqui proposta se baseia na heurística de sortear um vetor na vizinhança da solução pai do ES para ser o ponto inicial e, utilizando um algoritmo Quasi-Newton, encontrar um ótimo local da aproximação dentro de um raio, na região de busca.

3.5 Randomized KD-Tree

Dados um espaço métrico (aqui considera-se o Euclidiano) e um vetor de referência, a consulta de outros K vetores vizinhos armazenados em um banco de soluções avaliadas, com $N \geq K$ entradas, possui grande importância em aplicações nas áreas de visão computacional, aprendizado de máquina e mineração de dados, quando se consideram problemas de se gerar informação de caráter local relativa ao vetor de referência. Em tais aplicações, a estrutura de dados empregada para armazenamento da informação será determinante para a eficiência do acesso aos pontos pertencentes à vizinhança do vetor de referência.

Neste trabalho de tese, conforme Seção 3.4, deseja-se selecionar um subconjunto de soluções previamente avaliadas durante as iterações de um algoritmo evolutivo que serão utilizadas no processo de aproximação funcional na vizinhança de um ponto, para a execução de uma busca local ao redor desse ponto. Se acaso tal subconjunto representativo da vizinhança do ponto não fosse utilizada, uma forma de reunir uma amostra seria avaliar novas soluções na vizinhança do vetor de referência, como realizado em [da Cruz et al., 2011a]. Entretanto, a adoção do banco de soluções tem a vantagem de aproveitar o histórico da evolução dos valores de função-objetivo sem avaliações adicionais.

Em uma implementação do armazenamento da informação sem o uso de uma estrutura de dados espacial, em cada execução do procedimento de seleção dos pontos vizinhos pode-se calcular e armazenar em um vetor as N distâncias em relação ao ponto de referência em $O(N)$. Depois, o vetor contendo tais distâncias pode ser parcialmente ordenado a um custo $O(N \log N)$, de forma a permitir a determinação dos K vizinhos mais próximos. Uma desvantagem em tal processo é que sempre se faz necessário o cômputo das distâncias, em um contexto no qual o valor de N cresce à medida em que avançam as iterações do algoritmo evolutivo. Ademais, em tal procedimento haveria a característica determinística de se trazer exatamente os vizinhos mais próximos. Se porventura houvesse alguma pequena aleatoriedade na seleção de vizinhos, a aproximação funcional poderia variar um pouco, em problemas não lineares, e retornar soluções guiadas pelo histórico e ligeiramente diferentes, contribuindo assim para a diversificação da busca.

Para melhorar a eficiência de procedimentos de extração dos vizinhos mais próximos, diversas estruturas de dados em formato de árvore foram propostas ao longo do tempo. A *kd-tree* é uma das estruturas mais conhecidas para o armazenamento de informações a respeito de vizinhos mais próximos. Para a construção da árvore há um custo $O(nN \log N)$ e para a consulta dos K vizinhos há um custo esperado de $O(K \log N)$ em baixas dimensões [Friedman et al., 1977]. Porém, a performance deste procedimento sofre rápida degradação com o aumento da dimensão, havendo a necessidade de visitar muito mais ramificações da árvore durante a pesquisa, convergindo em direção a $O(KN)$.

Uma forma de se melhorar essa eficiência foi a criação das *randomized kd-trees*, que são capazes de realizar uma busca aproximada pelos K vizinhos de forma paralela em uma floresta. Tais *randomized kd-trees* que a compõem são construídas através de uma divisão espacial partindo primeiro das dimensões de maior variância, usando um componente aleatório [Muja & Lowe, 2009]. Com o uso das *randomized kd-trees* se torna possível o armazenamento e consulta de soluções previamente avaliadas que são

vizinhas mas não necessariamente aquelas mais próximas do ponto de referência, para aplicar no processo de aproximação funcional a um custo computacional satisfatório.

Para maiores detalhes técnicos sobre a *randomized kd-trees*, sugerem-se os artigos [Muja & Lowe, 2009] e [Muja & Lowe, 2014]. Outros trabalhos experimentais, que comparam a eficácia desta estrutura de dados em relação a outras ou que a utilizam em aplicações específicas com variações na implementação, podem ser encontradas nos trabalhos [Emiris & Nicolopoulos, 2013; Wan et al., 2014; Hutama et al., 2016; Xu et al., 2016; Hu et al., 2016].

3.6 Testes Estatísticos

Esta seção apresenta os testes estatísticos utilizados neste trabalho. Inicialmente é apresentado o teste-t de Student [Montgomery, 2012], aplicado no operador de comparação e reavaliação dinâmica, que é usado em duas das heurísticas propostas. Depois, apresenta-se o teste não paramétrico de Quade [Conover, 1999; García et al., 2010], que é usado para analisar a mediana dos dados gerados pelos experimentos desta tese.

3.6.1 Teste-t de Student

O teste-t de Student para duas amostras assume três condições [Montgomery, 2012]: (i) as variáveis aleatórias seguem uma distribuição Gaussiana; (ii) as variâncias destas são idênticas; e (iii) as duas amostras são independentes. A estatística de teste, apresentada pela Equação 3.5, é usada para testar a hipótese nula que afirma serem iguais as médias das amostras 1 e 2, em que n_1 e n_2 , \bar{x}_1 e \bar{x}_2 , e s_1^2 e s_2^2 , são o número de observações, a média amostral e a variância estimada das amostras 1 e 2, respectivamente.

$$t_0 = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}} \quad (3.5)$$

Para testar a hipótese nula, com um nível de significância $\alpha \in [0; 1]$, compara-se t_0 com a distribuição t com $n_1 + n_2 + 1$ graus de liberdade. Se $|t_0| > t_{\alpha/2, n_1 + n_2 + 1}$, então rejeita-se a hipótese nula, e aceita-se a alternativa que afirma que as médias são desiguais.

Este teste é usado para comparar os valores de função-objetivo ruidosa, provenientes de múltiplas avaliações de duas soluções distintas, durante as iterações das estratégias evolutivas que usam um operador que realiza a comparação das soluções considerando uma confiança estatística. Deseja-se que tal comparação melhore a de-

Tabela 3.1: Organização dos dados para o teste de Quade.

Bloco	Tratamento			
	1	2	...	t
1	$d_{1,1}$	$d_{1,2}$...	$d_{1,t}$
2	$d_{2,1}$	$d_{2,2}$...	$d_{2,t}$
⋮	⋮	⋮	⋮	⋮
b	$d_{b,1}$	$d_{b,2}$...	$d_{b,t}$

ção sobre qual é a melhor solução, quando se otimiza uma função-objetivo ruidosa. Este comportamento é esperado mesmo quando as amostras não respeitam a condição de seguirem uma distribuição Gaussiana, como ocorreu no trabalho [Syberfeldt et al., 2010].

3.6.2 Teste de Quade

O teste não paramétrico de Quade, proposto em [Quade, 1979], analisa a existência de diferença significativa entre as medianas de tratamentos em um projeto experimental com blocos completos e aleatorizados. Nesta tese, este teste é usado para analisar se há diferença entre as heurísticas investigadas nos diferentes experimentos.

Para realização do teste de Quade devem existir pelo menos $t \geq 2$ tratamentos organizados em b blocos, conforme organização apresentada pela Tabela 3.1, em que $d_{i,j}$ é a observação do i -ésimo bloco, $i \in \{1; 2; \dots; b\}$, para o j -ésimo tratamento, $j \in \{1; 2; \dots; t\}$. Além do mais, as seguintes hipóteses são assumidas: (i) cada informação de um bloco é obtida de forma independente; (ii) os dados podem ser ranqueados; e (iii) em cada bloco, os dados possuem um valor mínimo e máximo de modo que se possa determinar um intervalo entre eles [Conover, 1999].

Seja $r_{i,j}$ o ranque (colocação em uma ordenação crescente) assinalado ao dado $d_{i,j}$ dentro do próprio bloco. Se houver empate entre tratamentos, a média dos ranques é computada e atribuída para estes no bloco. Seja q_i o cômputo da diferença entre o maior e o menor valor observado no i -ésimo bloco. Considere $s_{i,j} = q_i(r_{i,j} - (t + 1)/2)$ e $s_j = \sum_{i=1}^b s_{i,j}$. Deste modo, a estatística f_0 , apresentada pela Equação 3.6, é usada para testar a hipótese nula que afirma que todos os tratamentos possuem o mesmo efeito, com um nível de significância $\alpha \in [0; 1]$.

$$f_0 = \frac{(b - 1)(1/b) \sum_{i=1}^t (s_j)^2}{\sum_{i=1}^b \sum_{j=1}^t (s_{i,j})^2 - (1/b) \sum_{i=1}^t (s_j)^2} \quad (3.6)$$

Isto é feito ao comparar com a distribuição T com $t - 1$ e $(b - 1)(t - 1)$ graus de

liberdade. Deste modo, se $f_0 > F_{(\alpha, t-1, (b-1)(t-1))}$ então a hipótese nula é rejeitada e considera-se que há pelo menos um par de tratamentos que se diferencia pela mediana.

Quando é detectada diferença significativa no interior de pelo menos um par de tratamentos, é de interesse identificar quais se diferenciam. Para isto, deve-se realizar um teste posterior que compara cada par de tratamentos. Aqui, para dois tratamentos distintos i e j , existe diferença se a condição da Equação 3.7, que usa a distribuição t , é satisfeita.

$$|s_i - s_j| < t_{(1-\alpha/2, (b-1)(t-1))} \sqrt{\frac{2b(\sum_{i=1}^b \sum_{j=1}^t (s_{i,j})^2 - (1/b) \sum_{i=1}^t (s_j)^2)}{(b-1)(t-1)}} \quad (3.7)$$

Salienta-se que a significância é ajustada através do procedimento de Holm [García et al., 2010] para reduzir o erro do tipo 1.

3.7 Algoritmos Propostos

3.7.1 ES-AP

A Figura 3.4 exibe um dos algoritmos propostos neste trabalho, denominado ES-AP (*Evolution Strategy with least squares APproximation*). Esse algoritmo utiliza uma busca local que (i) consulta uma amostra de soluções vizinhas previamente avaliadas, em uma *randomized kd-tree*; (ii) realiza uma aproximação linear ou quadrática, que é otimizada em uma região delimitada pelo raio σ da estratégia evolutiva, sendo o centro a solução pai; e (iii) retorna o ótimo aproximado para o algoritmo evolutivo como uma solução filho.

Além dos parâmetros originários do ES-BS, $p_u \in [0,8; 1,2]$, $p_a \in [5; 15]$ e $p_c \in [0,5; 1]$; há agora outros quatro parâmetros: $p_l \in [0,01; 0,5]$ representa a probabilidade de ocorrer a busca local em uma iteração; $p_\varepsilon \in [1e-4; 1]$ é usado no critério de parada e no cálculo do vetor gradiente, via diferenças finitas, para a rotina Quasi-Newton de minimização da função de aproximação; $p_{db} \in [1,1; 5]$ é o coeficiente do número de parâmetros da aproximação quadrática com hessiana simétrica cheia, $1 + (n^2 + 3n)/2$, para determinar o número máximo de soluções armazenadas em um banco de soluções; e $p_{sp} \in [1; 3]$ é o fator do número de parâmetros da função quadrática com hessiana simétrica cheia para determinar o número máximo de soluções vizinhas extraídas do banco para realizar a aproximação funcional.

Para se inicializar o ES-AP, definem-se o número de parâmetros da função linear, s_l , da função quadrática com hessiana diagonal, s_{dh} , e da função quadrática simétrica

```

Entrada:  $p_u, p_a, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}$ 
Dados:  $n, \mathbf{x}^{inf}, \mathbf{x}^{sup}, E_{max}, f$ 
1 início
2    $s_l \leftarrow n + 1; s_{dh} \leftarrow 2n + 1; s_{sh} \leftarrow 1 + (n^2 + 3n)/2;$ 
3    $db_{size} \leftarrow \lfloor p_{db} \cdot s_{sh} \rfloor; sp_{size} \leftarrow \lfloor p_{sp} \cdot s_{sh} \rfloor; u \leftarrow \lfloor p_u \cdot n \rfloor; a_{size} \leftarrow \lfloor p_a \cdot n \rfloor;$ 
4    $\sigma \leftarrow 1;$ 
5    $\mathbf{x}_P \leftarrow U(n, \mathbf{x}^{inf}, \mathbf{x}^{sup}); f_P \leftarrow f(\mathbf{x}_P); E \leftarrow 1;$ 
6    $B \leftarrow \emptyset; B.Insere((\mathbf{x}_P, f_P));$ 
7    $A \leftarrow \emptyset;$ 
8    $bl_{suc} \leftarrow \text{falso};$ 
9   enquanto  $E < E_{max}$  faça
10      $\mathbf{x}_C \leftarrow \mathbf{x}_P + \sigma \cdot [N_1(0, 1), \dots, N_n(0, 1)]'; f_C \leftarrow f(\mathbf{x}_C); E \leftarrow E + 1;$ 
11      $B.Insere((\mathbf{x}_C, f_C));$ 
12     se  $B.Tamanho() > db_{size}$  então  $B.RemovePrimeiro();$ 
13     se  $f_C < f_P$  então
14        $\mathbf{x}_P \leftarrow \mathbf{x}_C; f_P \leftarrow f_C; A.Insere(\text{sucesso});$ 
15     senão
16        $A.Insere(\text{falha});$ 
17     fim
18     se  $A.Tamanho() > a_{size}$  então  $A.RemovePrimeiro();$ 
19     se  $U(0, 1) < p_l \vee bl_{suc}$  então
20        $\tilde{f} \leftarrow \text{Aproxima}(\mathbf{x}_P, B, sp_{size}, s_l, s_{sh}, s_{sh});$ 
21       se  $\text{HouveAproximacao}(\tilde{f})$  então
22          $\mathbf{x}_L \leftarrow \text{QuasiNewtonApx}(\mathbf{x}_P, \tilde{f}, \mathbf{x}^{inf}, \mathbf{x}^{sup}, \sigma, p_\varepsilon);$ 
23          $f_L \leftarrow f(\mathbf{x}_L); E \leftarrow E + 1; B.Insere((\mathbf{x}_L, f_L));$ 
24         se  $B.Tamanho() > db_{size}$  então  $B.RemovePrimeiro();$ 
25         se  $f_L < f_P$  então
26            $\mathbf{x}_P \leftarrow \mathbf{x}_L; f_P \leftarrow f_L; A.Insere(\text{sucesso});$ 
27            $bl_{suc} \leftarrow \text{verdadeiro};$ 
28         senão  $A.Insere(\text{falha}); bl_{suc} \leftarrow \text{falso};$ 
29       fim
30     fim
31     se  $E \bmod u = 0$  então
32        $r_t \leftarrow A.TotalDe(\text{sucesso})/A.Tamanho();$ 
33        $\sigma \leftarrow \left\{ \begin{array}{ll} \sigma \cdot p_c & \text{se } r_t < 1/5 \\ \sigma/p_c & \text{se } r_t > 1/5 \\ \sigma & \text{se } r_t = 1/5 \end{array} \right\};$ 
34     fim
35   fim
36 retorna  $\mathbf{x}_P$ 

```

Figura 3.4: (1+1)-ES com busca local baseada na otimização de aproximação linear-quadrática designada por ES-AP.

cheia, s_{sh} . Decide-se a quantidade máxima de soluções avaliadas a serem armazenadas no banco de soluções (histórico avaliado) B , db_{size} , o tamanho máximo da amostra de vizinhos avaliados para realizar a aproximação funcional, sp_{size} , o número de avaliações consecutivas para atualizar o desvio σ , u , e o comprimento máximo do registro de sucessos e falhas das soluções geradas seguidamente, a_{size} . A seguir, estabelece-se o valor do desvio padrão a ser empregado e a primeira solução pai é gerada e avaliada. Tal progenitor é inserido com a respectiva avaliação em um par ordenado no banco de soluções avaliadas, previamente inicializada. A fila de sucessos e falhas começa vazia e a variável booleana bl_{suc} , que armazena a informação sobre o sucesso da busca local, recebe o valor falso.

Dentro do laço de repetição, uma solução filho é gerada, avaliada e armazenada no banco de pontos avaliados. Se esta for melhor que o pai atual, tal filho passa a assumir o posto parental. A informação sobre o sucesso do novo filho é arquivada na fila A . Depois, caso necessário, mantém-se o limite de a_{size} sucessos e falhas sucessivas, com a retirada do primeiro registro da fila. Em seguida, verifica-se se ocorrerá a busca local, que ocorre quando um número aleatório gerado a partir de uma distribuição uniforme é menor do que o valor de probabilidade, p_l , ou se a busca local executada na iteração anterior gerou uma solução melhor que o pai corrente daquela iteração (bl_{suc} é verdadeiro).

Se for ocorrer a busca local, então a função *Aproxima* receberá como parâmetros: um vetor centro na região de busca, \mathbf{x}_P , o banco de soluções previamente avaliadas, B , a quantidade máxima de amostras que serão usadas para se gerar a aproximação, sp_{size} , e os números de parâmetros das funções linear, s_l , quadrática com hessiana diagonal, s_{dh} , e simétrica, s_{sh} . Inicialmente, um conjunto de até sp_{size} vizinhos mais próximos e distintos de \mathbf{x}_P será selecionado a partir de B . Se porventura a quantidade de pares ordenados (solução e avaliação) extraída de B for menor ou igual a p_l , então não ocorrerá aproximação funcional. Se for maior que p_l e menor ou igual a p_{dh} , \tilde{f} aproximará linearmente a função f . Se for maior que p_{dh} e menor ou igual a p_{sh} , ocorrerá a aproximação por uma função quadrática com hessiana diagonal. Para uma situação com mais de p_{sh} amostras, a aproximação será feita por uma função quadrática com hessiana simétrica arbitrária. Em qualquer caso, utiliza-se a formulação apresentada na Equação 3.3.

O próximo passo é verificar o resultado da tentativa de aproximação funcional. Em caso negativo, prossegue-se para as operações restantes da estratégia evolutiva. Em caso positivo, executa-se o procedimento *QuasiNewtonAprx*, ilustrado pela Figura 3.5, que possui os seguintes parâmetros: um vetor centro, \mathbf{x} , a função aproximada, \tilde{f} , os limites inferiores e superiores, \mathbf{x}^{inf} e \mathbf{x}^{sup} , um raio r que delimita o ótimo da aproxi-

```

Entrada:  $\mathbf{x}, \tilde{f}, \mathbf{x}^{inf}, \mathbf{x}^{sup}, r, p_\epsilon$ 
Dados:  $n$ 
1 início
2    $ls_{qtd} \leftarrow \min(\lfloor r \cdot n \rfloor, n);$ 
3    $\tilde{f}_L \leftarrow \infty;$ 
4   para  $i \in \{1, \dots, ls_{qtd}\}$  faça
5      $\mathbf{x}_{tmp} \leftarrow [\mathbf{x}[1] + U(-r, r), \dots, \mathbf{x}[n] + U(-r, r)]';$ 
6      $\tilde{f}_{tmp} \leftarrow \tilde{f}(\mathbf{x}_{tmp});$ 
7      $[\mathbf{x}_{tmp}, \tilde{f}_{tmp}] \leftarrow QuasiNewton2(\mathbf{x}_{tmp}, \tilde{f}_{tmp}, \tilde{f}, \mathbf{x}^{inf}, \mathbf{x}^{sup}, p_\epsilon, \mathbf{x}, r);$ 
8     se  $\tilde{f}_{tmp} < \tilde{f}_L$  então
9        $\mathbf{x}_L \leftarrow \mathbf{x}_{tmp};$ 
10       $\tilde{f}_L \leftarrow \tilde{f}_{tmp};$ 
11    fim
12  fim
13 fim
14 retorna  $\mathbf{x}_L$ 

```

Figura 3.5: Função *QuasiNewtonApx* de otimização da aproximação linear-quadrática.

mação em uma hipersfera cujo centro é \mathbf{x} , e um limiar p_ϵ usado na implementação do algoritmo Quasi-Newton para otimizar a aproximação no critério de parada e cômputo do gradiente (conforme feito na Figura 3.3).

Nos casos em que \tilde{f} é uma quadrática, há a possibilidade de haver diversos ótimos locais sobre a hipersfera de centro \mathbf{x} e raio r . Por isso, para otimizar a aproximação, é executado um algoritmo Quasi-Newton ls_{qtd} vezes, número este proporcional ao tamanho do raio e dimensão do problema, com diferentes pontos iniciais. Uma diferença desta versão em relação àquela apresentada pela Figura 3.3 é que esta última também trata a restrição do ótimo estar dentro da hipersfera de centro \mathbf{x} e raio r . Ademais, como se otimiza a aproximação, não há gasto com avaliações de função-objetivo durante esta busca local. As iterações deste Quasi-Newton terminam com os mesmos critérios usados pela versão da Figura 3.3. No final de *QuasiNewtonApx*, a melhor solução encontrada de acordo com \tilde{f} nas ls_{qtd} iterações é retornada para o algoritmo evolutivo. Tal solução obtida via busca local é avaliada e comparada com a solução pai. Se for melhor, então esta assume o posto do pai e um “sucesso” é enfileirado. Caso contrário, uma “falha” é registrada.

Com o fluxo da aplicação de volta para a estratégia evolutiva, a última ação no laço de repetição diz respeito a atualização de σ pela regra do 1/5. Estas operações se repetem até que o máximo de avaliações de função-objetivo seja atingido. Ao final, a melhor solução encontrada, \mathbf{x}_P , é retornada.

```

Entrada:  $p_u, p_a, p_c, p_{uE}, p_\alpha$ 
Dados:  $n, \mathbf{x}^{inf}, \mathbf{x}^{sup}, E_{max}, f$ 
1 início
2    $u \leftarrow \lfloor p_u \cdot n \rfloor; \quad a_{size} \leftarrow \lfloor p_a \cdot n \rfloor;$ 
3    $\mathbf{x}_P \leftarrow U(n, \mathbf{x}^{inf}, \mathbf{x}^{sup}); \quad F_P \leftarrow \emptyset;$ 
4    $f_P \leftarrow f(\mathbf{x}_P); \quad F_P.Insere(f_P);$  /* 2x */
5    $E \leftarrow 2; \quad uE \leftarrow 2;$ 
6    $\sigma \leftarrow 1; \quad A \leftarrow \emptyset;$ 
7   enquanto  $E < E_{max}$  faça
8      $\mathbf{x}_C \leftarrow \mathbf{x}_P + \sigma \cdot [N_1(0, 1), \dots, N_n(0, 1)]'; \quad F_C \leftarrow \emptyset;$ 
9      $f_C \leftarrow f(\mathbf{x}_C); \quad F_C.Insere(f_C);$  /* 2x */
10     $E \leftarrow E + 2; \quad uE \leftarrow 1 + \lceil p_{uE} \cdot E / E_{max} \rceil;$ 
11     $[c, E] \leftarrow Compara(\mathbf{x}_C, F_C, \mathbf{x}_P, F_P, p_\alpha, f, uE, E, E_{max});$ 
12    se  $c = "C \stackrel{p_\alpha}{<} P"$  então
13       $\mathbf{x}_P \leftarrow \mathbf{x}_C; \quad F_P \leftarrow F_C; \quad A.Insere(\text{sucesso});$ 
14    senão  $A.Insere(\text{falha});$ 
15    se  $A.Tamanho() > a_{size}$  então  $A.RemovePrimeiro();$ 
16    se  $E \bmod u = 0$  então
17       $r_t \leftarrow A.TotalDe(\text{sucesso}) / A.Tamanho();$ 
18       $\sigma \leftarrow \left\{ \begin{array}{ll} \sigma \cdot p_c & \text{se } r_t < 1/5 \\ \sigma / p_c & \text{se } r_t > 1/5 \\ \sigma & \text{se } r_t = 1/5 \end{array} \right\};$ 
19    fim
20  fim
21 fim
22 retorna  $\mathbf{x}_P$ 

```

Figura 3.6: (1+1)-ES com operador de comparação com confiança estatística designada por ES-CC.

3.7.2 ES-CC

Outra heurística projetada nesta tese, ES-CC (*Evolution Strategy with Comparison based on statistical Confidence*), exibida na Figura 3.6, é uma versão do ES-BS com o operador de comparação baseado no teste-t de Student para duas amostras [Montgomery, 2012] e reavaliação dinâmica, voltada exclusivamente para otimizar objetivos ruidosos. Esta implementação possui cinco parâmetros que são os três presentes no ES-BS, $p_u \in [0,8; 1,2]$, $p_a \in [5; 15]$ e $p_c \in [0,5; 1]$; e mais os outros dois que são: $p_{uE} \in \{2; \dots; 10\}$, o número máximo de reavaliações para uma única solução, no final da execução; e $p_\alpha \in [0,6; 0,9]$, o nível de confiança adotado no teste-t pelo operador de comparação (ou seja, o nível de significância é $1 - p_\alpha$).

No ES-CC, cada solução é avaliada pelo menos duas vezes em relação a função-

objetivo. O número de reavaliações é limitado por uma quantidade que cresce linearmente com o avanço das avaliações global até um limiar predefinido. A qualidade de uma solução é dada pela amostra avaliada de objetivo. No operador proposto, para comparar os valores de função-objetivo avaliados de duas soluções, utiliza-se o teste-t de Student para duas amostras. Se houver diferença significativa, então a heurística prossegue usando a informação obtida. Se não foi detectada diferença, então aquela solução de maior variância na amostra avaliada é reamostrada, desde que não tenha ainda atingido o máximo permitido de reavaliações. Caso este limite já tenha sido alcançado, então a outra solução deve ser reavaliada caso não tenha atingido o máximo permitido de avaliações por solução. Se ambas soluções já tiverem atingido o máximo de reavaliações, então a comparação é feita usando-se a média amostral. Tal operador visa aumentar o grau de certeza na realização de operações de tomada de decisão no interior do algoritmo que envolvam a comparação entre valores de função-objetivo em pontos distintos. Almeja-se, assim, empregar um maior esforço computacional no momento em que as soluções já se encontram evoluídas, quando é de se esperar que haja maior semelhança entre as soluções sendo consideradas.

Inicialmente, no algoritmo, define-se o número de avaliações consecutivas para se atualizar o raio σ , u , e o tamanho da fila de sucessos/falhas, a_{size} . Gera-se a solução pai, \mathbf{x}_P , cujo conjunto de avaliações, F_P , é inicializado. Na linha seguinte, o progenitor é avaliado duas vezes e tais valores de objetivo são armazenados. Em seguida, o número corrente de avaliações global, E , a quantidade máxima de reavaliações por solução, uE , o desvio padrão para geração de filhos (raio), σ , e a fila de sucessos/falhas, A , são inicializados.

No laço de repetição, que dura enquanto o número de avaliações global não atingir um máximo, ocorre o processo evolutivo. Gera-se a solução filho, que é avaliada duas vezes inicialmente. Atualiza-se o total de avaliações global, E , e o máximo de reavaliações permitidas por uma única solução, uE .

A comparação baseada em confiança entre o pai e o filho ocorre em seguida na função *Compara*, delineada na Figura 3.7, que recebe como parâmetros um par de soluções com as respectivas avaliações (por referência), o nível de confiança, a função-objetivo ruidosa, o número máximo de avaliações permitida por solução, a quantidade global de avaliações de função-objetivo já realizadas e o total de avaliações que o algoritmo pode avaliar. De início, supõe-se que ambas soluções são iguais até que se mostre o contrário. No laço, que é executado apenas quando está disponível a possibilidade de realizar novas avaliações de função-objetivo, realiza-se o teste-t com significância $1 - p_\alpha$ entre os conjuntos avaliados. Como resultado da comparação, pode-se retornar “ $i \stackrel{p_\alpha}{<} j$ ”, “ $i \stackrel{p_\alpha}{>} j$ ” ou “ $i \stackrel{p_\alpha}{=} j$ ”, se i é estatisticamente menor, ou maior que j ,

```

Entrada:  $\mathbf{x}_i, F_i, \mathbf{x}_j, F_j, p_\alpha, f, uE, E, E_{max}$ 
1 início
2    $c \leftarrow "i \stackrel{p_\alpha}{=} j";$ 
3   enquanto  $E < E_{max}$  faça
4      $c \leftarrow \text{TesteT}(F_i, F_j, p_\alpha);$ 
5     se  $(c \in \{ "i \stackrel{p_\alpha}{<} j", "i \stackrel{p_\alpha}{>} j" \}) \vee (F_i.E = uE \wedge F_j.E = uE)$  então
6       saia do enquanto;
7     fim
8     se  $(\text{Desvio}(F_i) > \text{Desvio}(F_j) \wedge F_i.E < uE) \vee F_j.E = uE$  então
9        $f_i \leftarrow f(\mathbf{x}_i); F_i.\text{Insere}(f_i); B.\text{Insere}((\mathbf{x}_i, f_i));$ 
10      senão
11         $f_j \leftarrow f(\mathbf{x}_j); F_j.\text{Insere}(f_j); B.\text{Insere}((\mathbf{x}_j, f_j));$ 
12      fim
13       $E \leftarrow E + 1;$ 
14    fim
15    se  $c = "i \stackrel{p_\alpha}{=} j"$  então
16      se  $\text{Media}(F_i) < \text{Media}(F_j)$  então  $c \leftarrow "i \stackrel{p_\alpha}{<} j";$ 
17      senão  $c \leftarrow "i \stackrel{p_\alpha}{>} j";$ 
18    fim
19 fim
20 retorna  $[c, E]$ 

```

Figura 3.7: Função *Compara* que confronta duas soluções com base no teste-t e realiza reavaliações para reduzir o desvio.

ou não existe diferença significativa entre ambas, respectivamente. Se ambas soluções foram amostradas até o máximo permitido ou se foi detectada a diferença, então aborta-se o laço de repetição. Caso contrário, avalia-se aquela solução que apresenta o maior desvio padrão e que não tenha atingido o número máximo de avaliações por solução. Se ambas soluções forem avaliadas até o máximo permitido e ainda não tiver sido detectada diferença, então a média amostral é usada para determinar o resultado da comparação. Também é retornado o número global de avaliações atualizado.

Este procedimento é utilizado para decidir se a diferença entre as amostras de valores de objetivos provenientes de duas soluções distintas é significativa ou não, sendo eventualmente recomendada nova reavaliação de soluções neste último caso.

De volta, no escopo do algoritmo evolutivo, se o filho for melhor que o pai, ocorre a substituição deste por aquele. A informação sobre o sucesso ou falha desse filho é enfileirada em A . Por último, verifica-se se é o momento de se aplicar a regra do 1/5 para atualização do raio σ . Em caso positivo, isto é feito.

Esta heurística opera segundo princípios que são parecidos com os que são empre-

gados em diversos algoritmos propostos na literatura para o tratamento de problemas de otimização de funções ruidosas [Aizawa & Wah, 1994; Miller, 1997; Jin & Branke, 2005; Siegmund et al., 2013; Li et al., 2014; Akimoto et al., 2015; Rakshit et al., 2016; Merele et al., 2016].

3.7.3 ES-APCC

A última heurística proposta nesta parte da tese, ES-APCC (*Evolution Strategy with least squares APproximation and Comparison based on statistical Confidence*), é mostrada na Figura 3.8. Esta heurística é uma combinação do ES-AP e ES-CC, que também é especificamente orientada para problemas de otimização ruidosos. Dessa forma, como no ES-CC, soluções já avaliadas são reavaliadas e comparadas pelo operador que usa o teste-t [Montgomery, 2012], conforme algoritmo delineado na Figura 3.7. Tais avaliações são usadas na busca local para gerar a aproximação linear-quadrática que é otimizada pelo algoritmo da Figura 3.5, assim como é feito no ES-AP. Com a combinação destes dois operadores, espera-se um bom aproveitamento na busca global e local pelas soluções dos problemas.

No ES-APCC há nove parâmetros, sendo os sete primeiros provenientes do ES-AP e os dois últimos adicionados ao ES-CC, que são: $p_u \in [0,8; 1,2]$, $p_a \in [5; 15]$, $p_c \in [0,5; 1]$, $p_l \in [0,01; 0,5]$, $p_\varepsilon \in [1e-4; 1]$, $p_{db} \in [1,1; 5]$, $p_{sp} \in [1; 3]$, $p_{uE} \in \{2; \dots; 10\}$, e $p_\alpha \in [0,6; 0,9]$.

A inicialização do ES-APCC é similar à do ES-AP. Em adição a tal processo, inicializa-se a solução pai, a estrutura que armazena o conjunto de avaliações ruidosas do progenitor e o banco de soluções avaliadas. Em seguida e em duas vezes, avalia-se, armazena-se a avaliação do pai em F_P e insere-se tal amostra no banco. Com isso, armazena-se o número inicial de avaliações global e de quantidade máxima de computações de objetivo por solução. Depois, define-se o tamanho inicial do raio, inicializa-se a fila de sucessos e supõe-se que não ocorrerá busca local na primeira iteração.

No laço de repetição, que termina quando o número de avaliações atinge um limite prefixado, a solução filho é gerada e a respectiva estrutura, F_C , que armazena os objetivos é inicializada. Avalia-se este vetor duas vezes, arquivam-se as avaliações do objetivo ruidoso e insere-se essa informação no banco. Logo após, atualiza-se a quantidade global de avaliações e o máximo de repetições de avaliação de objetivo por solução, uE . Antes da comparação entre as soluções pai e filho, há uma análise sobre a possível atualização da quantidade de amostras avaliadas a serem armazenadas.

A comparação baseada em confiança entre o pai e o filho ocorre em seguida na

```

Entrada:  $p_u, p_a, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}, p_{uE}, p_\alpha$ 
Dados:  $n, \mathbf{x}^{inf}, \mathbf{x}^{sup}, E_{max}, f$ 
1 início
2    $s_l \leftarrow n + 1; s_{dh} \leftarrow 2n + 1; s_{sh} \leftarrow 1 + (n^2 + 3n)/2;$ 
3    $db_{size} \leftarrow \lfloor p_{db} \cdot s_{sh} \rfloor; sp_{size} \leftarrow \lfloor p_{sp} \cdot s_{sh} \rfloor; u \leftarrow \lfloor p_u \cdot n \rfloor; a_{size} \leftarrow \lfloor p_a \cdot n \rfloor;$ 
4    $\mathbf{x}_P \leftarrow U(n, \mathbf{x}^{inf}, \mathbf{x}^{sup}); F_P \leftarrow \emptyset; B \leftarrow \emptyset;$ 
5    $f_P \leftarrow f(\mathbf{x}_P); F_P.Insere(f_P); B.Insere((\mathbf{x}_P, f_P));$  /* 2x */
6    $E \leftarrow 2; uE \leftarrow 2;$ 
7    $\sigma \leftarrow 1; A \leftarrow \emptyset; bl_{suc} \leftarrow \text{falso};$ 
8   enquanto  $E < E_{max}$  faça
9      $\mathbf{x}_C \leftarrow \mathbf{x}_P + \sigma \cdot [N_1(0, 1), \dots, N_n(0, 1)]'; F_C \leftarrow \emptyset;$ 
10     $f_C \leftarrow f(\mathbf{x}_C); F_C.Insere(f_C); B.Insere((\mathbf{x}_C, f_C));$  /* 2x */
11     $E \leftarrow E + 2; uE \leftarrow 1 + \lceil p_{uE} \cdot E / E_{max} \rceil;$ 
12    se  $B.Tamanho() > db_{size}$  então  $B.RemovePrimeiro();$ 
13     $[c, E] \leftarrow Comparar(\mathbf{x}_C, F_C, \mathbf{x}_P, F_P, p_\alpha, f, uE, E, E_{max});$ 
14    se  $c = "C \stackrel{p_\alpha}{<} P"$  então
15       $\mathbf{x}_P \leftarrow \mathbf{x}_C; F_P \leftarrow F_C; A.Insere(\text{sucesso});$ 
16    senão  $A.Insere(\text{falha});$ 
17    se  $A.Tamanho() > a_{size}$  então  $A.RemovePrimeiro();$ 
18    se  $U(0, 1) < p_l \vee bl_{suc}$  então
19       $\tilde{f} \leftarrow Aproxima(\mathbf{x}_P, B, sp_{size}, s_l, s_{sh}, s_{sh});$ 
20      se  $HouveAproximacao(\tilde{f})$  então
21         $\mathbf{x}_L \leftarrow QuasiNewtonApx(\mathbf{x}_P, \tilde{f}, \mathbf{x}^{inf}, \mathbf{x}^{sup}, \sigma, p_\varepsilon); F_L \leftarrow \emptyset;$ 
22         $f_L \leftarrow f(\mathbf{x}_L); F_L.Insere(f_L); B.Insere((\mathbf{x}_L, f_L));$  /* 2x */
23         $E \leftarrow E + 2; uE \leftarrow 1 + \lceil p_{uE} \cdot E / E_{max} \rceil;$ 
24        se  $B.Tamanho() > db_{size}$  então  $B.RemovePrimeiro();$ 
25         $[c, E] \leftarrow Comparar(\mathbf{x}_L, F_L, \mathbf{x}_P, F_P, p_\alpha, f, uE, E, E_{max});$ 
26        se  $c = "L \stackrel{p_\alpha}{<} P"$  então
27           $\mathbf{x}_P \leftarrow \mathbf{x}_L; F_P \leftarrow F_L; A.Insere(\text{sucesso});$ 
28           $bl_{suc} \leftarrow \text{verdadeiro};$ 
29          senão  $A.Insere(\text{falha}); bl_{suc} \leftarrow \text{falso};$ 
30      fim
31    fim
32    se  $E \bmod u = 0$  então
33       $r_t \leftarrow A.TotalDe(\text{sucesso}) / A.Tamanho();$ 
34       $\sigma \leftarrow \left\{ \begin{array}{ll} \sigma \cdot p_c & \text{se } r_t < 1/5 \\ \sigma / p_c & \text{se } r_t > 1/5 \\ \sigma & \text{se } r_t = 1/5 \end{array} \right\};$ 
35    fim
36  fim
37 retorna  $\mathbf{x}_P$ 

```

Figura 3.8: (1+1)-ES com busca local baseada na otimização de aproximação linear-quadrática e comparação com confiança estatística designada por ES-APCC.

função *Compara*, conforme Figura 3.7, da mesma forma como ocorre no ES-CC. Se o filho for melhor que o pai, então ocorre a substituição. O sucesso ou a falha do filho é enfileirada em *A*. Em seguida, utilizando uma regra similar à empregada no ES-AP, pode ocorrer a busca local baseada na otimização da aproximação funcional. A solução retornada pela busca é avaliada, armazenada e comparada com o pai atual, da mesma forma que ocorre com as soluções-filho. Depois de tais etapas, prossegue-se com os passos finais do laço de repetição desta estratégia evolutiva.

3.7.4 Propriedades dos Operadores Propostos

Foram propostos dois novos operadores, um de busca local com função de aproximação de baixa ordem (linear ou quadrática) e outro de reamostragem da função ruidosa. Esses operadores foram concebidos visando produzir os seguintes efeitos, no contexto da otimização de funções ruidosas:

busca local: O operador de busca local baseado em aproximação por funções de baixa ordem deve produzir implicitamente uma filtragem do ruído, com a redução do efeito deste sobre o resultado da busca. Seria de se esperar que uma função aproximante normalmente produzisse efeitos de filtragem, desde que o número de parâmetros de tal função fosse menor que a quantidade de amostras utilizadas para o ajuste da função aproximada. Neste trabalho, são empregadas especificamente funções de baixa ordem (funções lineares ou quadráticas), de forma que espera-se que tal efeito de filtragem seja mais pronunciado do que seria no caso de funções de mais alta ordem. Deve-se mencionar que os trabalhos de Zhou et al. [2003]; Granichin [2004] e Févotte et al. [2008] aplicaram regressão linear para filtrar ruídos de diferentes naturezas em aplicações práticas diversas, explorando o mesmo princípio a ser utilizado aqui.

reamostragem: A operação de reamostragem, por sua vez, visa reduzir a proporção da tomada de decisões equivocadas no interior do algoritmo em decorrência do ruído. O operador de comparação baseado no teste-t com reavaliações dinâmicas tem a função de reduzir a chance de que se escolha uma solução pior que outra em uma comparação pareada. Isto ocorre devido ao uso do teste estatístico para a comparação das duas soluções, dado um par de amostras avaliadas, com um nível de confiança fornecido. Com isto, espera-se que a convergência do algoritmo ES seja mais rápida utilizando tal operação do que ocorreria em um cenário sem a utilização de tal operador. Esse princípio foi também empregado, de maneira diversa daquela empregada aqui, no trabalho de Syberfeldt et al. [2010], que

otimizou modelos multiobjetivo com funções-objetivo ruidosas, cujas avaliações contêm ruídos aditivos provenientes de diferentes distribuições.

Deve-se notar que, embora a proposição do operador de busca local por funções aproximantes de baixa ordem tenha motivação ligada ao tratamento de problemas de otimização com funções ruidosas, tal operador pode potencialmente favorecer também a otimização de funções não ruidosas, uma vez que essa busca local pode acelerar a convergência de um algoritmo evolutivo pela simples utilização de uma representação local para a geometria da função-objetivo. Por este motivo, é construída uma versão de algoritmo ES, o ES-AP, incluindo apenas esse operador de busca local, o qual será testado no próximo capítulo também no contexto da otimização de funções isentas de ruído. O algoritmo ES-CC faz uso do operador de reamostragem dinâmica, que é específico para o contexto de funções ruidosas. O algoritmo ES-APCC, por sua vez, inclui os dois operadores propostos, e também somente faz sentido no contexto de funções ruidosas.

O próximo capítulo desta tese apresenta os resultados de testes realizados com os algoritmos propostos, em comparações realizadas com a versão básica do algoritmo ES (o ES-BS) e com uma versão de algoritmo ES hibridizada com uma busca local baseada no algoritmo Quasi-Newton (o ES-QN).

Capítulo 4

Resultados

Este capítulo apresenta os resultados experimentais que visam validar os algoritmos propostos nesta tese. Os experimentos encontram-se organizados em três grupos:

- Experimentos no contexto de funções não ruidosas, que visam comparar o algoritmo ES-AP com um algoritmo ES básico (ES-BS) e com um algoritmo ES com busca local baseada em método Quasi-Newton (ES-QN).
- Experimentos que realizam a comparação dos três algoritmos anteriormente examinados com os algoritmos ES-CC e ES-APCC, no contexto de funções ruidosas.
- Experimentos no contexto de um problema ruidoso motivado em uma aplicação prática, no domínio da epidemiologia matemática.

Nos experimentos serão utilizadas as seguintes ferramentas de análise de resultados:

- Gráficos do tipo beanplot [Kampstra et al., 2008; Phillips, 2016] que representam os dados brutos segundo uma forma geométrica que indica a densidade dos valores na distribuição, superposta a uma barra grossa maior indicando o valor da mediana, uma caixa cinza transparente destacando os dados do primeiro ao terceiro quartil, e uma barra entre o quantil 0,05 e 0,95.
- Tabelas com as estatísticas de mínimo, primeiro quartil, mediana, terceiro quartil e máximo.
- Teste estatístico não paramétrico de Quade [Conover, 1999] com o procedimento de Holm para ajustar o nível de significância na comparação pareada [García et al., 2010].
- Para o primeiro e segundo experimentos, em relação a todas as respectivas execuções das estratégias evolutivas, são apresentados gráficos que descrevem a evo-

lução da mediana do melhor valor de função-objetivo obtido pelo número de avaliações computado.

Todos os experimentos foram realizados com a codificação dos algoritmos em C++11 [Stroustrup, 2013], utilizando as bibliotecas Eigen 3 [Guennebaud et al., 2010], GNU Scientific Library [Gough, 2009] e ALGLIB [Bochkanov, 2016], sendo a compilação realizada no g++ 5.4.1 com otimização O3. Os gráficos e análise dos dados foram gerados no R-project [Matloff, 2011]. O equipamento utilizado foi um computador com processador Intel Core I7, 16 GB de RAM, VGA 4 GB DDR 3 GFORCE GT 630 no Ubuntu 16.04, sem Internet e dedicado para os testes.

4.1 Experimento 1 - Comparação dos Algoritmos em Objetivos Não Ruidosos

O primeiro experimento aqui apresentado diz respeito à otimização de funções não-ruidosas. Nesse experimento, o algoritmo ES-AP é comparado com uma estratégia evolutiva simples, ES-BS, e com uma estratégia evolutiva com busca local do tipo Quasi-Newton, ES-QN. Os algoritmos ES-BS, ES-QN e ES-AP, com parâmetros sintonizados [Eiben & Smit, 2011] pela heurística RPS, de acordo com o Apêndice A, são utilizados para realizar a otimização das 12 funções-objetivo [Liang et al., 2013] mencionadas na Tabela 4.1 na dimensão 10, com região de busca $[-100; 100]$ para cada variável, e ajustadas para que no ótimo o valor da função-objetivo seja igual a 0 (zero). Menciona-se também que cada função-objetivo é rotacionada por uma matriz particular e possui a solução ótima localizada em diferentes posições, não sendo o centro da região de busca.

Cada algoritmo dispõe de um orçamento de 1000 avaliações de função-objetivo por execução, sendo executado 30 vezes para cada função-objetivo, com distintas sementes para o gerador de números aleatórios. Isso significa que cada algoritmo será representado por 360 resultados, sendo 30 resultados para cada uma das 12 funções-objetivo. O resultado, para cada heurística, é representado pelo melhor valor de função-objetivo encontrado durante a execução. Os algoritmos são executados, neste experimento, com os parâmetros sintonizados conforme mostrado na Tabela 4.2.

O nível de significância igual a 0,05 é utilizado em um teste de hipótese cuja hipótese nula afirma que as medianas das estimativas do valor ótimo da função-objetivo encontradas pelos algoritmos, considerando todas as execuções, são iguais. A hipótese alternativa desse teste afirma que pelo menos um dos algoritmos resulta em mediana diferente das demais.

Tabela 4.1: Conjunto de funções-objetivo com os valores estimados de amplitude no valor de objetivo em um raio de distância 1 do ótimo.

Função	Amplitude	Características
Elliptic	$3,7193 \cdot 10^6$	Unimodal; não-separável; quadrática mal-condicionada.
Bent Cigar	$1,5784 \cdot 10^7$	Unimodal; não-separável; suave porém estreita.
Discus	$2,0545 \cdot 10^6$	Unimodal; não-separável; com uma direção sensível.
Rosenbrock	5,0863	Multimodal; não-separável; possui um vale estreito do ótimo local para o global.
Ackley	6,5721	Multimodal; não-separável.
Weierstrass	2,4268	Multimodal; não-separável; contínua, mas diferenciável somente em um conjunto de pontos.
Griewank	1,2362	Multimodal; não-separável.
Rastrigin	9,2487	Multimodal; não-separável; quantidade alta de ótimos locais.
Schwefel	207,0700	Multimodal; não-separável; quantidade alta de ótimos locais, cujo segundo melhor é longe do ótimo global.
Katsuura	6,2114	Multimodal; não-separável; contínua, mas não diferenciável.
HappyCat	0,9613	Multimodal; não-separável.
HGBat	3,2226	Multimodal; não-separável.

Tabela 4.2: Parâmetros sintonizados para objetivos não ruidosos.

Heurística	Parâmetros
ES-BS	$(p_u, p_s, p_c) = (0,856; 9,035; 0,674)$
ES-QN	$(p_u, p_s, p_c, p_l, p_\varepsilon) = (1,038; 14,506; 0,712; 0,129; 0,068)$
ES-AP	$(p_u, p_s, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}) = (0,885; 14,080; 0,888; 0,271; 0,009; 2,511; 1,475)$

Os 360 valores obtidos por cada uma das heurísticas ES-BS, ES-QN e ES-AP, nas 30 execuções para cada uma das 12 funções-objetivo, são representados na Figura 4.1. Neste beanplot observa-se que, com um mesmo orçamento de avaliação, a distribuição da amostra gerada pelo algoritmo proposto ES-AP atinge valores de função-objetivo menores, quando comparados àqueles produzidos pelos outros algoritmos estudados. Verifica-se nessa figura que o ES-AP obteve valores de função-objetivo em geral menores que os correspondentes quantis dos outros algoritmos, além de apresentar distribuição de valores com menor dispersão. Deve-se notar ainda que os gráficos de beanplot dos diferentes algoritmos exibem agrupamentos de valores de função-objetivo que permitem identificar, nesses gráficos que agregam todas as funções-objetivo, as contribuições de diferentes funções – o que permite inferir o comportamento de cada algoritmo em cada função. O detalhamento dos resultados, com a exibição dos gráficos correspondentes a cada função-objetivo, é apresentado no Apêndice B.

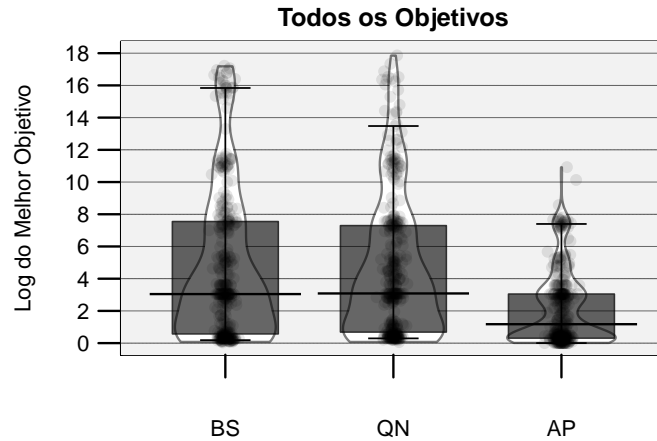


Figura 4.1: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação a todas as funções-objetivo.

Tabela 4.3: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação a todas as funções-objetivo sem ruído.

Estatística	BS	QN	AP
Mínimo	0,0397	0,0610	0,0000
1º Quartil	0,5726	0,6867	0,3104
Mediana	3,0445	3,0839	1,1730
3º Quartil	7,5454	7,2941	3,0445
Máximo	17,2193	17,8617	10,9260

Tabela 4.4: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação a todas as funções-objetivo sem ruído.

	ES-BS	ES-QN
ES-QN	0,448097	–
ES-AP	0,000000	0,000000
Valor-p do Teste: 0,000000		

O exame da Tabela 4.3 permite concluir que o algoritmo ES-AP atingiu o valor ótimo para pelo menos alguma função-objetivo, e alcançou o melhor valor, dentre todos os algoritmos, para cada um dos quantis examinados.

A Tabela 4.4 mostra o valor-p obtido no teste de Quade, que indica a existência de diferença significativa entre, pelo menos, algum par de algoritmos em relação aos melhores valores de objetivo retornados nas execuções. Isso está de acordo com a diferença detectada entre o ES-AP e as outras duas estratégias pelo valor-p menor que 10^{-6} . Por outro lado, não se observou diferença significativa entre os algoritmos ES-BS e ES-QN.

Para observar o comportamento da mediana do valor de objetivo (considerando

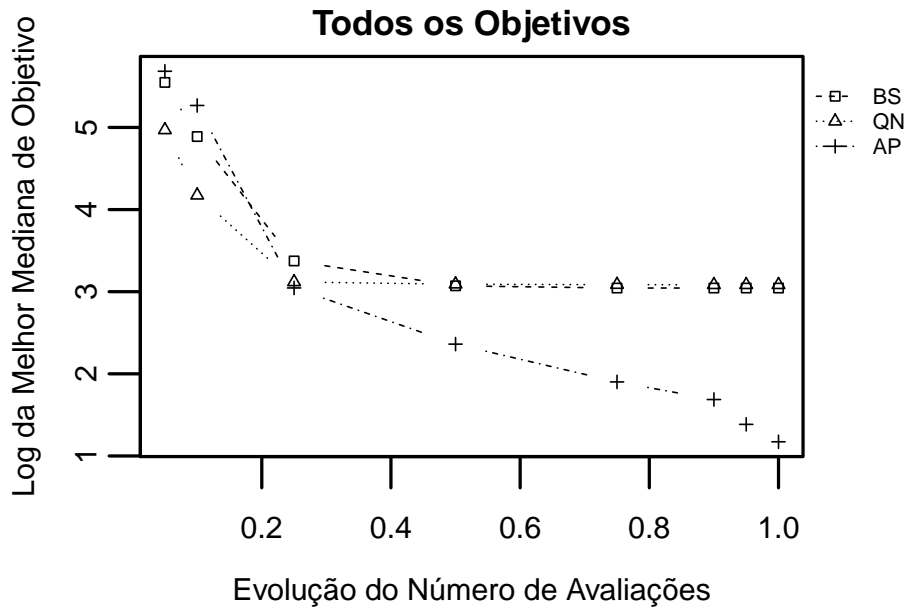


Figura 4.2: Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo obtido pelos algoritmos pela progressão do número de avaliações, em relação a todas as funções-objetivo.

as 30 execuções de cada algoritmo) da melhor solução como função do número de avaliações de função-objetivo, construiu-se a Figura 4.2. Nesse gráfico é possível observar como a busca local baseada na otimização de uma aproximação de função contribuiu para a melhoria da convergência. Nota-se que no início a heurística ES-QN gera soluções melhores até vinte e cinco por cento do total de avaliações e, a seguir, permanece estagnada com soluções cuja qualidade é próxima daquelas geradas pela heurística ES-BS. Por outro lado, após esse número de avaliações, a heurística ES-AP ultrapassa o ES-QN, sendo que a diferença do valor da função-objetivo alcançado persiste aumentando até o final da execução.

Com base nesses resultados, pode-se afirmar que a heurística ES-AP foi superior em relação às heurísticas ES-BS e ES-QN para a otimização do conjunto considerado de funções não-ruídas.

A análise detalhada dos dados considerados é apresentada no Apêndice B, em que as Figuras B.1 e B.2 e Tabelas B.1 e B.2, apresentam os resultados separadamente para cada função-objetivo. Tais saídas se encontram sintetizadas na Tabela 4.5.

O exame da Tabela 4.5 permite concluir que a estratégia ES-AP parece globalmente superior às outras duas, só não tendo obtido o primeiro lugar no ranking para uma das funções-objetivo. Mesmo nesse caso, o ES-AP não foi significativamente pior que o ES-BS, que por sua vez não foi também significativamente pior que o ES-QN.

Tabela 4.5: Ranking da mediana do valor da função-objetivo para as diferentes heurísticas, considerando cada função-objetivo, para problemas não-ruídos. Considera-se empate no caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas heurísticas no ranking 1, a terceira heurística recebe ranking 2 se ficar empatada com uma das duas primeiras, ou ranking 3 se for significativamente diferente das duas primeiras.

	ES-BS	ES-QN	ES-AP
Elliptic	3	2	1
Bent Cigar	3	2	1
Discus	2	2	1
Rosenbrock	1	2	1
Ackley	1	3	1
Weierstrass	1	1	1
Griewank	2	3	1
Rastrigin	2	1	1
Schwefel	1	1	1
Katsuura	1	1	1
HappyCat	1	1	2
HGBat	1	1	1

Alguns aspectos interessantes podem ser observados nessa tabela. Por exemplo, nota-se que em funções unimodais convexas, como a Elliptic e a Bent Cigar, os métodos ES-AP e ES-QN superam o método ES-BS, o que seria de se esperar, dado que esses fazem uso de aproximações quadráticas da função-objetivo. Não haveria uma razão a priori, nesses casos, nem para um melhor desempenho do ES-AP em relação à ES-QN nem para o contrário disso. Os experimentos revelaram, no entanto, o melhor desempenho do ES-AP. No caso de funções unimodais não convexas, como a Rosenbrock, ou de funções multimodais, como a Rastrigin, seria de se esperar que as aproximações quadráticas empregadas tanto pelo ES-AP quanto pelo ES-QN pudessem não representar tamanha vantagem, o que de fato se verificou, tendo o ES-BS nos dois casos atingido o primeiro lugar no ranking, ao lado do ES-AP. Nesses casos, pode-se inferir que a forma de utilização da aproximação quadrática empregada no ES-AP pelo menos não prejudicou o desempenho do algoritmo, ao contrário da formulação empregada no ES-QN, que causou degradação no desempenho do algoritmo. Pode-se conjecturar que essa diferença ocorra pelo fato de que o ES-QN utiliza a informação da aproximação quadrática de maneira estritamente local, perfazendo uma busca em linha, enquanto o ES-AP utiliza informação semelhante de uma maneira semi-global, que pode permitir que a aproximação realize uma “filtragem” na paisagem de *fitness* da função, auxiliando a heurística na fuga de pontos de mínimo local.

Tabela 4.6: Parâmetros sintonizados para objetivos ruidosos.

Heurística	Parâmetros
ES-BS	$(p_u, p_s, p_c) = (0,926; 7,531; 0,757)$
ES-QN	$(p_u, p_s, p_c, p_l, p_\varepsilon) = (0,841; 12,860; 0,505; 0,444; 0,345)$
ES-AP	$(p_u, p_s, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}) = (1,143; 14,508; 0,659; 0,114; 0,009; 2,973; 1,817)$
ES-CC	$(p_u, p_s, p_c, p_E, p_\alpha) = (0,805; 6,231; 0,899; 2; 0,727)$
ES-APCC	$(p_u, p_s, p_c, p_E, p_\alpha, p_l, p_\varepsilon, p_{db}, p_{sp}) = (0,806; 5,786; 0,563; 2; 0,856; 0,368; 0,338; 2,967; 2,033)$

4.2 Experimento 2 - Comparação dos Algoritmos em Objetivos Ruidosos

O segundo experimento a ser apresentado neste capítulo diz respeito à situação em que funções ruidosas devam ser otimizadas. Neste experimento, as heurísticas ES-AP, ES-CC e ES-APCC são comparadas entre si e com as heurísticas ES-BS e ES-QN, sobre as mesmas funções-objetivo empregadas no teste anterior, agora sujeitas a ruído Gaussiano aditivo, sendo que 8 diferentes níveis de ruído são utilizados nos testes. Cada heurística é executada 30 vezes para cada uma das 12 funções-objetivo considerando cada um dos 8 níveis de ruído, sendo que a cada execução a heurística dispõe de um orçamento de 1000 avaliações de função-objetivo. Cada nível de desvio experimentado vale 10^{-i} multiplicado pela amplitude do valor de objetivo na proximidade do ótimo, conforme Tabela 4.1, para todo $i \in \{1, \dots, 8\}$ e cada função-objetivo.

Cada uma das heurísticas foi previamente submetida a um procedimento de ajuste de parâmetros específico para a situação de funções ruidosas, conforme descrito no Apêndice A. Os parâmetros obtidos de tal ajuste são mostrados na Tabela 4.6.

Uma síntese dos resultados de todo esse segundo experimento é exposta na Figura 4.3. Constata-se neste beanplot que o ES-AP, de um modo geral, retornou soluções melhores e mais robustas. Tal constatação também pode ser feita a partir da Tabela 4.7, que mostra as magnitudes das diferenças favoráveis ao ES-AP em relação a todas as demais heurísticas, observadas em todos os quantis.

Tomando-se as estatísticas do ES-AP como base e subtraindo as estatísticas das outras heurísticas, tem-se as seguintes diferenças do $\log(f(x) + 1)$ para o valor de função-objetivo para o mínimo, primeiro, segundo e terceiro quartis, e máximo. Em relação ao ES-BS as diferenças foram 0, $-0,1618$, $-0,0289$, $-2,3825$ e $-4,2759$. Isto sinaliza que (i) ambas conseguiram atingir o ótimo para alguma função-objetivo, (ii) o ES-AP foi relativamente melhor do que o ES-BS em relação ao primeiro quartil e mediana, e (iii) o ES-AP foi muito melhor ao reduzir o terceiro quartil e o objetivo máximo produzido. Em comparação ao ES-QN, as respectivas diferenças foram $-0,0557$, $-0,6359$, $-0,6626$, $-3,0150$ e $-5,0074$, o que indica uma crescente superioridade do

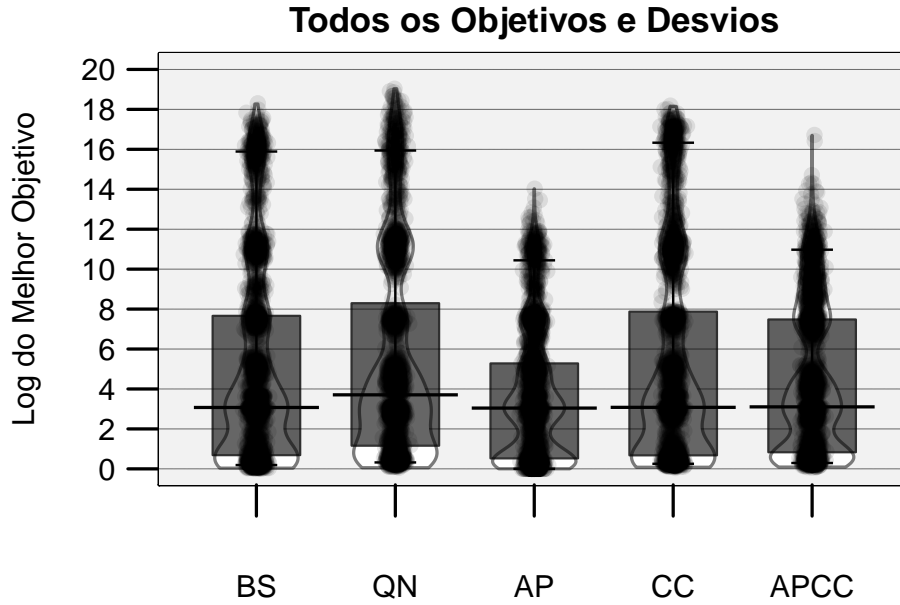


Figura 4.3: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.

Tabela 4.7: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.

Estatística	ES-BS	ES-QN	ES-AP	ES-CC	ES-APCC
Mínimo	0,0000	0,0557	0,0000	0,0426	0,0639
1° Quartil	0,6872	1,1613	0,5254	0,6780	0,8319
Mediana	3,0734	3,7071	3,0445	3,0810	3,1049
3° Quartil	7,6607	8,2932	5,2782	7,8710	7,4785
Máximo	18,3034	19,0349	14,0275	18,1859	16,7200

ES-AP, do valor de objetivo mínimo até o máximo retornado. No que concerne à ES-CC, as diferenças foram $-0,0426$, $-0,1526$, $-0,0365$, $-2,5928$ e $-4,1584$, assinalando uma relativa proximidade do menor valor até a mediana, e posteriormente ampliando a desigualdade. Por fim, com referência ao ES-APCC, as diferenças foram $-0,0639$, $-0,3065$, $-0,0604$, $-2,2003$ e $-2,6925$, que revelam uma comparação parecida com a do ES-CC, porém com uma variação menor para os valores maiores.

O resultado do teste de Quade sobre a igualdade das medianas dos valores de objetivos reais deste experimento é exposto na Tabela 4.8. É indicada a existência de diferença significativa em relação a qualquer par de algoritmos, e a ordem, da menor para a maior mediana, é dada por: ES-AP, ES-BS, ES-CC, ES-APCC e ES-QN.

A evolução da mediana do $\log(f(x) + 1)$ para o valor de melhor objetivo, de

Tabela 4.8: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,000000	0,000000	–	–
CC	0,000000	0,000000	0,000000	–
APCC	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000				

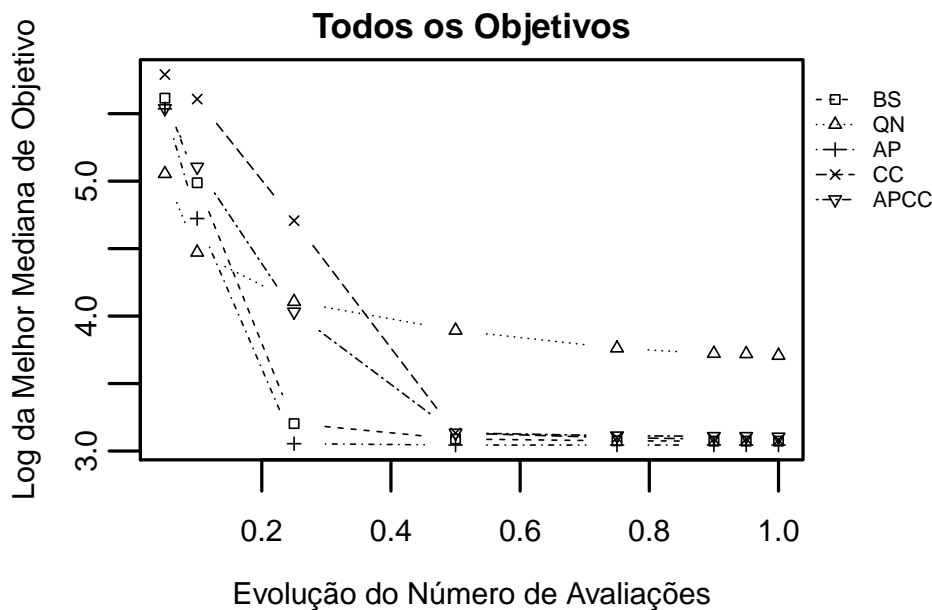
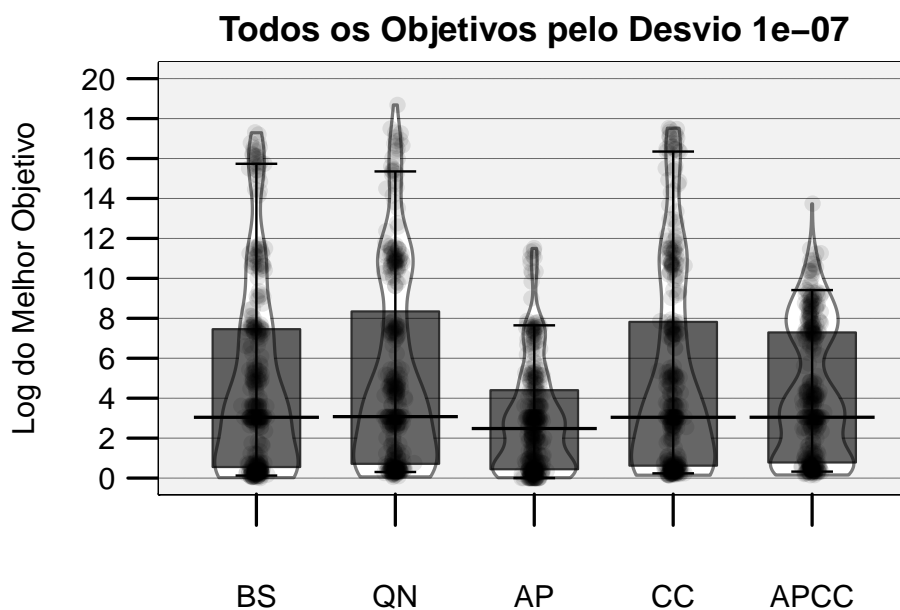
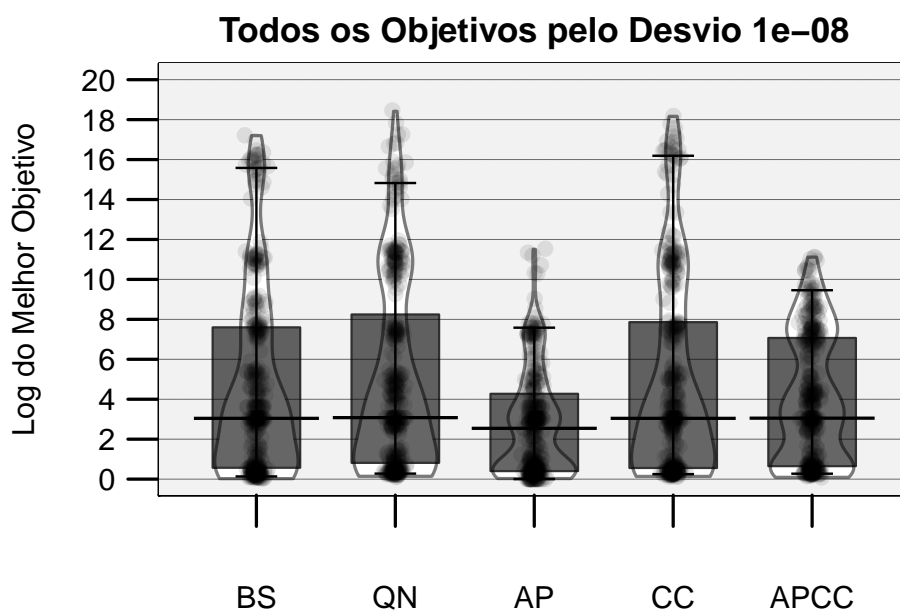


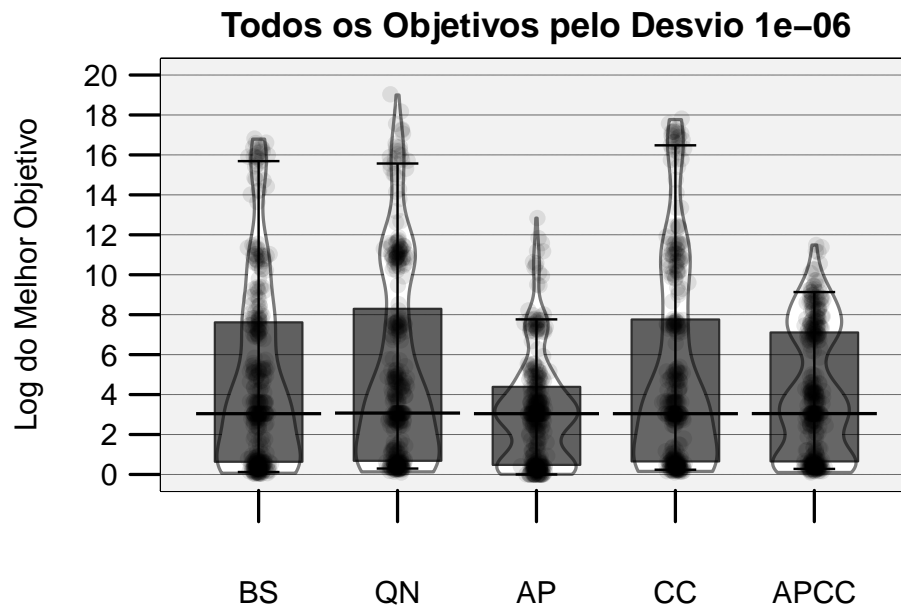
Figura 4.4: Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo real obtido pelos algoritmos pela progressão do número de avaliações, em relação a todas as funções-objetivo ruidosas e todos os níveis de ruído.

acordo com o número de avaliações, é mostrada na Figura 4.4. Nesse gráfico observa-se que, a partir de vinte e cinco por cento do total de avaliações, a versão ES-AP se tornou a melhor estratégia e o ES-BS a segunda melhor. As estratégias ES-CC e ES-APCC, respectivamente em terceira e quarta posições ao final, estagnaram após a metade do total de avaliações. Isto ocorreu devido às reavaliações de soluções únicas, que causaram um dispêndio no orçamento de avaliações de função-objetivo que não foi compensado com uma melhoria da convergência. O ES-QN chegou a ser a melhor heurística nos primeiros dez por cento de avaliações. Porém, depois disto, a taxa de melhoria a cada passo foi baixa, ocorrendo estagnação após setenta e cinco por cento do total de objetivos computados – resultando em mediana bem pior do que as das outras heurísticas.

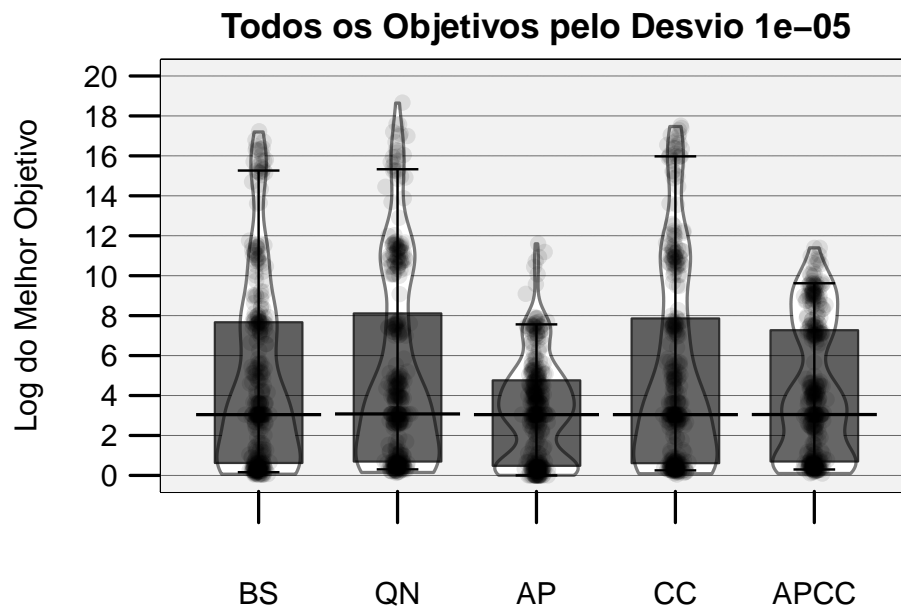
Prosseguindo, a Figura 4.5 exhibe os beanplots com o $\log(f(x) + 1)$ dos melhores

valores de objetivo real retornado para cada estratégia por nível de desvio padrão. Tal gráfico revela como se comporta a dificuldade de otimização com o incremento do desvio-padrão do ruído. Nota-se para todos os níveis de ruído que as soluções retornadas pelos métodos ES-AP e ES-APCC são melhores, em praticamente todos os quantis, que as soluções produzidas pelo algoritmo básico ES-BS, sendo que este apresenta soluções melhores que aquelas produzidas pelos algoritmos ES-CC e ES-QN. O melhor algoritmo, em todos os casos, foi o ES-AP.

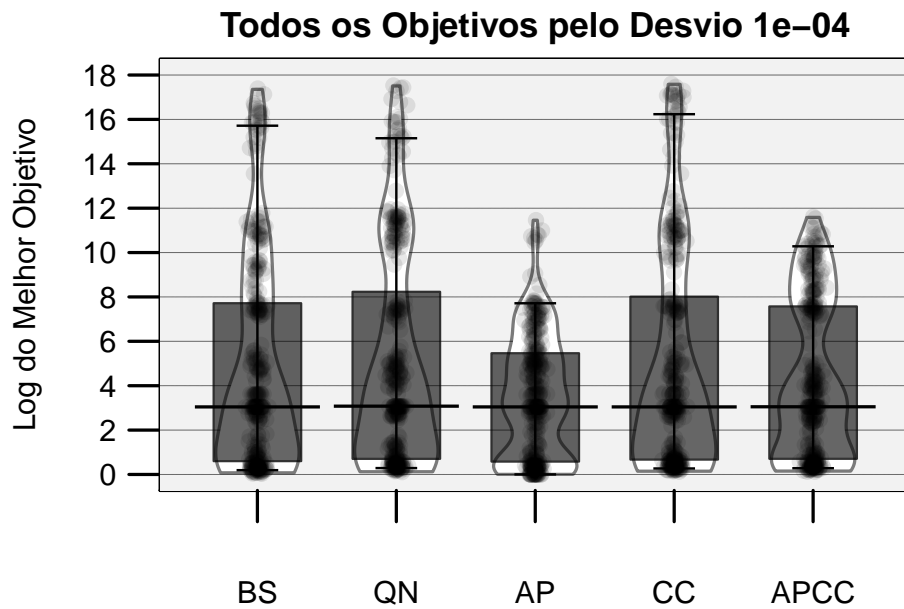
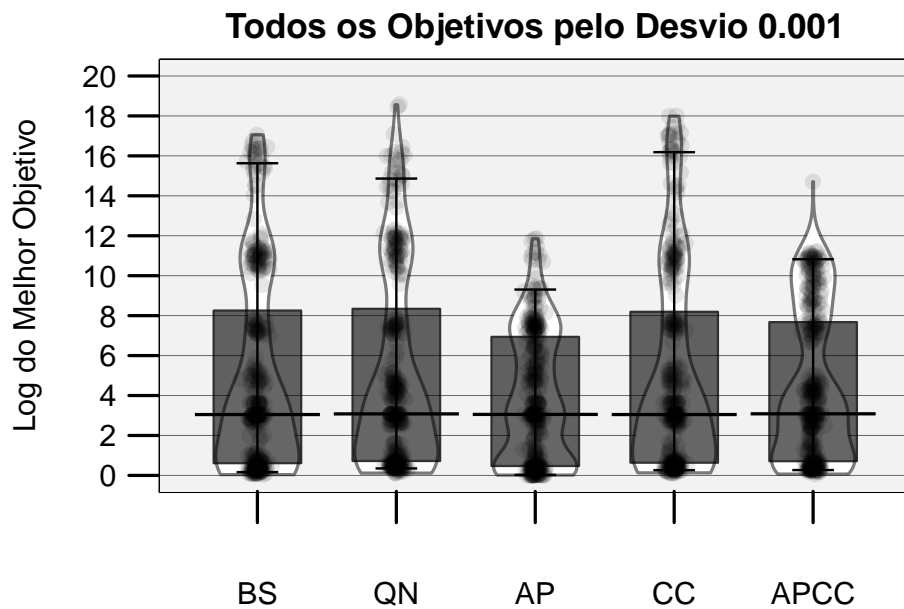




(c) Desvio de $1e-6$.



(d) Desvio de $1e-5$.

(e) Desvio de $1e-4$.(f) Desvio de $1e-3$.

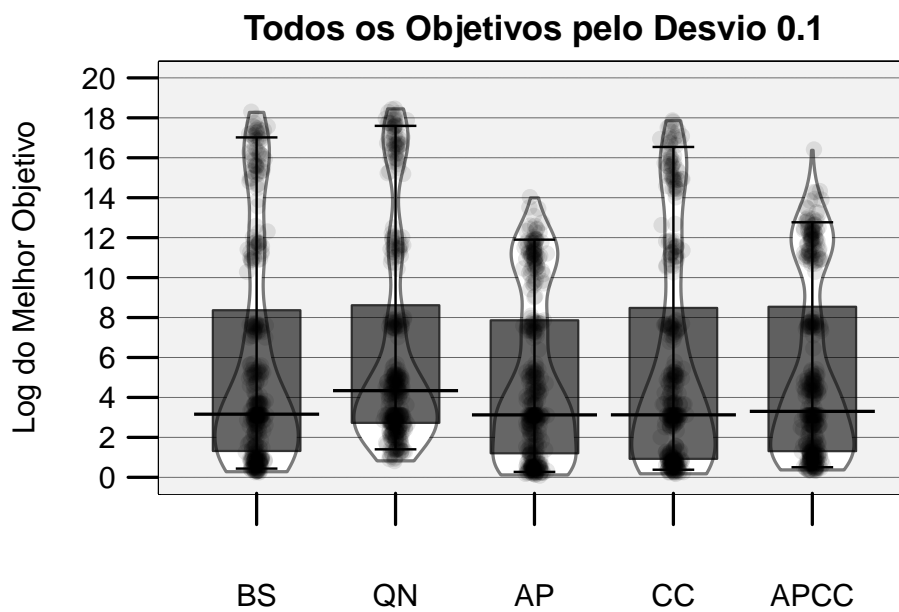
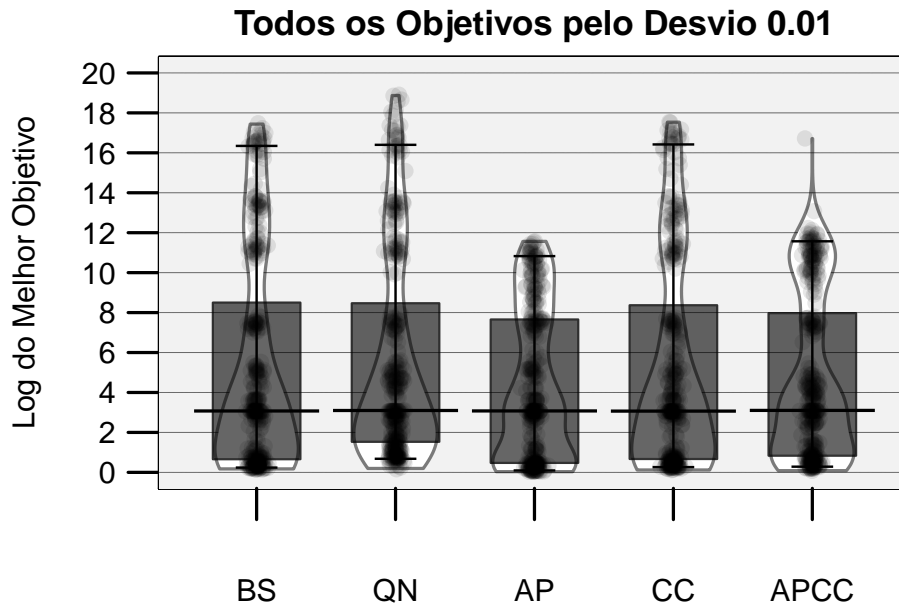


Figura 4.5: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivos reais retornados pelos algoritmos em relação a todas as funções-objetivo em cada nível de ruído.

O detalhamento dos resultados obtidos para cada função-objetivo é apresentado nas Tabelas B.3 e B.4 do Apêndice B. Esses resultados são sintetizados na Tabela 4.9. Quanto à metodologia para construção desta tabela, deve-se mencionar que o caráter não transitivo da relação de equivalência neste contexto faz com que não seja imediata a construção de um ranking a partir das relações de equivalência ou diferença entre os diversos elementos de um conjunto. Assim, mesmo se não há diferença significativa entre A e B e também não há diferença significativa entre B e C , ainda pode ser que exista diferença significativa entre A e C . A construção dos rankings exibidos na Tabela 4.9 adota então o seguinte critério:

- (i) Faz-se $k \leftarrow 1$;
- (ii) O algoritmo de melhor mediana recebe ranking k ;
- (iii) Caso haja p algoritmos cujas medianas não apresentem diferença significativa em relação ao algoritmo anterior, todos esses também recebem ranking k ;
- (iv) O primeiro algoritmo que não tiver mediana equivalente à do algoritmo indicado no passo (ii) recebe ranking $k + 1$ caso tenha mediana não-significativamente diferente que algum ou alguns dos algoritmos indicados no passo (iii), ou ranking $(k + p + 1)$ caso tenha mediana significativamente diferente de todos esses;
- (v) Caso ainda existam algoritmos não ranqueados, faz-se $k \leftarrow k + p + 1$, e retorna-se ao passo (iii).

Os resultados obtidos neste experimento sugerem conclusões interessantes:

- A simples introdução da reavaliação de soluções para aumentar a confiança nas escolhas feitas por um algoritmo ES não foi vantajosa, em geral, neste experimento, não tendo havido ganho de convergência associado ao custo adicional de avaliações de função. Desta forma a heurística ES-CC foi, em geral, pior que o algoritmo básico ES-BS.
- No contexto de funções ruidosas, a heurística ES-QN parece ter tido desempenho um pouco degradado em relação ao contexto isento de ruídos. Isso pode ser medido pela comparação com a heurística básica ES-BS: o ES-QN foi melhor ou igual à heurística ES-BS por 8 vezes, no caso sem ruído, e apenas 6 vezes no caso com ruído.
- A heurística ES-AP, por outro lado, havia sido melhor ou igual à heurística ES-BS por 11 vezes no contexto isento de ruído, passando a ficar melhor ou igual à ES-BS por 12 vezes (a totalidade dos casos) na situação com ruído.

Tabela 4.9: Ranking da mediana do valor da função-objetivo para as diferentes heurísticas, considerando cada função-objetivo, agregando-se todos os níveis de ruído. Considera-se empate no caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas ou mais heurísticas em determinado ranking, uma heurística situada logo abaixo recebe ranking um a mais se ficar empatada com uma das anteriores, ou ranking igual ao número de heurísticas anteriores mais um se for significativamente diferente das anteriores. A entrada na tabela indicada com (*) indica heurística não equivalente às outras em decorrência da existência de expressivo número de *outliers*.

	ES-BS	ES-QN	ES-AP	ES-CC	ES-APCC
Elliptic	3	3	1	5	2
Bent Cigar	2	5	1	4	2
Discus	3	3	1	3	2
Rosenbrock	1	5	1	1	4
Ackley	1	5	1	1	4
Weierstrass	3	2	2	3	1
Griewank	2	5	1	3	4
Rastrigin	3	2	3	3	1
Schwefel	1	1	1	1	1
Katsuura	2	5	1	1	5
HappyCat	5	2	1	5	5
HGBat	1	(*)	1	1	1

- Nas duas únicas funções para as quais a heurística ES-AP não é a melhor, no contexto de funções ruidosas, a primeira colocação no ranking fica com a heurística ES-APCC. Ou seja, em todos os casos, a heurística de melhor desempenho faz uso do procedimento proposto de aproximação quadrática.
- A heurística ES-APCC apresentou desempenho melhor ou igual ao da heurística básica ES-BS para oito funções no contexto de funções ruidosas, o que a situa em posição melhor que as heurísticas ES-BS, ES-CC e ES-QN.

Em síntese, os resultados obtidos parecem sugerir, para o contexto estudado, que a maneira mais eficaz de tratar problemas de otimização de funções ruidosas seria pela aplicação de aproximações lineares-quadráticas como mecanismo gerador de novas soluções. Pode-se conjecturar que essas aproximações quadráticas produzam um efeito de filtragem do ruído, desta forma produzindo uma melhoria da convergência do algoritmo.

O mecanismo mais comumente empregado na literatura para tratar ruído, que é o da reamostragem, não se mostrou eficaz nos testes realizados quando aplicado isoladamente. Quando aplicada em conjunto com a aproximação quadrática, a reamostragem

produziu efeito positivo, mas o melhor custo-benefício foi alcançado pela simples aplicação da aproximação, que parece fazer o melhor uso de um orçamento limitado de avaliações de função.

Uma possível explicação para essa observação seria a de que deve ser mais vantajoso examinar um novo ponto, ainda não avaliado, do que repetir a avaliação de um ponto já observado. O possível ganho que seria obtido com essa repetição da avaliação do mesmo ponto já é aproximadamente obtido pelo efeito de filtragem causado pela aproximação quadrática, que produz novos “palpites” sobre os valores de função-objetivo que já agregam a informação de várias amostragens (embora não do mesmo ponto). Em outras palavras, o efeito de redução do ruído obtido com a reamostragem do mesmo ponto também é alcançado com a utilização das diferentes amostras de pontos distintos na síntese da função de aproximação quadrática. Esta amostragem de pontos diferentes tem a vantagem adicional de trazer informação nova, sobre pontos anteriormente não amostrados. Tal observação parece constituir uma contribuição relevante desta tese ao conhecimento hoje existente.

4.3 Experimento 3 - Comparação dos Algoritmos em um Problema de Controle de Epidemia

Um terceiro experimento é agora apresentado considerando um problema com motivação prática que se enquadra na categoria de problemas ruidosos: o controle de epidemias através de políticas de vacinação. Neste experimento, as heurísticas ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC são comparadas, sendo utilizados os mesmos conjuntos de parâmetros que foram sintonizados para ambiente ruidoso conforme descritos no segundo experimento.

Uma epidemia é definida como a ocorrência de casos de uma doença em quantidade superior àquela normalmente esperada em uma população. Já uma endemia corresponde a um padrão relativamente estável de ocorrência de uma doença em um grupo populacional, com uma prevalência relativamente alta nesse grupo. Um grande desafio em saúde pública está relacionado ao planejamento de campanhas de vacinação com o objetivo de erradicar uma doença ou de controlar a sua propagação, evitando a ocorrência de grandes picos de epidemia ou estados endêmicos com alta prevalência, considerando a limitação dos recursos disponíveis [Kim-Farley, 1992].

No trabalho de pesquisa desenvolvido no âmbito desta tese (ver artigo publicado no Apêndice C), foi desenvolvida uma formulação multiobjetivo que considera um compromisso entre custo e efetividade para determinar políticas não-dominadas e

robustas para o controle de uma doença através de vacinação impulsiva (concentrada em determinados momentos). Uma solução é não-dominada se ela não é pior que qualquer outra em termos tanto de custo quanto de efetividade, e uma solução robusta não tem seu desempenho degradado de maneira significativa quando diferentes realizações dos processos estocásticos subjacentes são consideradas. O compromisso entre a efetividade das políticas de controle das epidemias e o custo de sua implementação já foi reconhecida em alguns trabalhos, os quais empregaram abordagens de otimização multiobjetivo para a síntese de políticas de vacinação. O presente trabalho de pesquisa, desenvolvido de acordo com tal abordagem, também apresenta as seguintes contribuições metodológicas específicas:

- (i) Todo processo epidêmico apresentará dois regimes dinâmicos distintos: um regime transitório e um outro permanente. Se, por um lado, a fase transitória requer um controle variante no tempo para tratar de maneira ótima o sistema, por outro lado a fase permanente será mais adequadamente tratada por um controle também em regime permanente, uma vez que a síntese de tal controle poderá ser realizada de maneira mais precisa por meio de uma formulação com menor número de graus de liberdade. Neste trabalho, a política de controle é construída como uma concatenação de uma sequência de ações de controle sintetizada para a fase transitória, seguida por uma sequência invariante no tempo. Isto permite uma melhor eficiência da política de controle. A metodologia usual de promover uma única política de vacinação ao longo de toda a duração do processo epidêmico leva a políticas que podem se encontrar distantes da otimalidade [Kim-Farley, 1992; Hinman, 1999; Hethcote, 2000]. Em comparação com trabalhos anteriores, a abordagem aqui apresentada adiciona novos graus de liberdade ao problema, permitindo diferentes tamanhos de pulsos de controle em distintos instantes, além de permitir a aplicação de pulsos em momentos de tempo arbitrários nas duas fases. Dado um horizonte de tempo, uma política de controle é representada aqui pelo número de campanhas de vacinação, pelos instantes de tempo em que cada uma será executada, e pelo número de indivíduos que devem ser imunizados em cada campanha.
- (ii) A síntese de políticas de vacinação é usualmente realizada, na grande maioria dos trabalhos que já abordaram o assunto, utilizando um modelo de epidemia baseado em equações diferenciais que representa o comportamento médio da epidemia na população, tal como o modelo endêmico Suscetível-Infetado-Recuperado (SIR) [Kermack & McKendrick, 1933; Anderson & May, 1992; Hethcote, 2000]. A simulação deste modelo é computacionalmente pouco dispendiosa, o que faz com

que este tipo de modelo seja adequado para utilização dentro de um algoritmo de otimização no qual o modelo é avaliado um grande número de vezes. No entanto, como o comportamento real da epidemia em uma população de tamanho finito será caracterizada por variações estocásticas nas variáveis de interesse, uma simulação que leve em conta tais efeitos se torna necessária para realizar uma avaliação mais realística do resultado da aplicação da política de vacinação. Um Modelo Baseado em Indivíduos (MBI) [Grimm & Railsback, 2005] que corresponde em média à representação por equações diferenciais é aqui empregado para avaliar o conjunto de políticas que emergem do procedimento de otimização multiobjetivo, de forma a avaliar a sensibilidade de cada política à estocasticidade subjacente ao processo. Este modelo é também empregado para estimar alguns índices de desempenho da política de vacinação, tais como a probabilidade de erradicação da doença. Nenhum trabalho anteriormente publicado apresentou alternativas para tratar tais problemas, de forma que tal tratamento pode ser considerado uma contribuição deste trabalho.

Devido à característica indicada no item (ii), o problema aqui tratado é dotado de função-objetivo ruidosa, sendo portanto adequado para ser tratado por meio de algoritmos tais como o ES-AP e o ES-APCC. O restante deste capítulo é dedicado a apresentar especificamente a questão da comparação entre as abordagens ES-AP, ES-APCC, ES-CC, ES-QN e ES-BS sobre tal problema de otimização ruidoso – o que constitui apenas parte do trabalho de pesquisa que foi desenvolvido sobre a questão da síntese de políticas de vacinação. Uma apresentação mais detalhada do procedimento que foi desenvolvido para a síntese de políticas de vacinação, conforme descrito acima, é exibida no artigo [da Cruz et al., 2017], que é resultante do trabalho de pesquisa desenvolvido no âmbito desta tese (ver artigo publicado no Apêndice C).

4.3.1 Modelos de Epidemias

4.3.1.1 Modelo SIR

O modelo SIR descreve a dinâmica dos indivíduos susceptíveis, infectados e recuperados em uma população durante a evolução de uma epidemia, de maneira média. Esse modelo pode ser usado para descrever o contágio por vírus que possam ser transmitidos entre indivíduos, tais como sarampo ou rubéola. O vírus infecta indivíduos susceptíveis que se tornam infectados. Estes se recuperam após algum tempo, tornando-se imunes. Quando existe uma vacina e esta é aplicada a um indivíduo susceptível, este também se torna “recuperado”, sem ter passado pelo estágio de “infectado” [Hethcote, 2000].

O modelo SIR utiliza uma estratégia de compartimentos que se relacionam por um sistema de três equações diferenciais. O problema de valor inicial é apresentado na Equação (4.1). As variáveis S , I e R representam, respectivamente, o número de indivíduos susceptíveis, infectados e recuperados. O termo N representa o número de indivíduos que é suposto constante: $S(t) + I(t) + R(t) = N$, $\forall t \geq 0$. Os parâmetros são a taxa de transmissão β , a taxa de recuperação de indivíduos infectados γ , e a taxa de perda da imunidade por parte dos indivíduos recuperados μ . As unidades de medida desses parâmetros são o inverso da unidade de tempo. Os valores $1/\gamma$ e $1/\mu$ representam o tempo médio de infecção e o tempo médio até a perda da imunidade de um indivíduo, respectivamente.

$$\begin{aligned} \frac{dS}{dt} &= \mu N - \mu S - \frac{\beta IS}{N}, & S(0) &\geq 0, \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I - \mu I, & I(0) &\geq 0, \\ \frac{dR}{dt} &= \gamma I - \mu R, & R(0) &\geq 0, \end{aligned} \tag{4.1}$$

Dividindo-se a Equação (4.1) por N , o sistema fica expresso em termos da fração de indivíduos susceptíveis, s , e infectados, i . Isto leva à Equação (4.2). Portanto, a fração recuperada é computada como $r(t) = 1 - s(t) - i(t)$.

$$\begin{aligned} \frac{ds}{dt} &= \mu - \mu s - \beta is, & s(0) &\geq 0, \\ \frac{di}{dt} &= \beta is - \gamma i - \mu i, & i(0) &\geq 0, \end{aligned} \tag{4.2}$$

O número básico de reprodução R_0 para o modelo SIR, definido pela Equação (4.3), representa o número médio de infecções produzidas por um indivíduo infectado. O sistema SIR possui um estado de equilíbrio endêmico assintoticamente estável se e somente se $R_0 > 1$, o que é mostrado na Equação (4.4) [Hethcote, 2000].

$$R_0 = \frac{\beta}{\mu + \gamma} \tag{4.3}$$

$$s_e = \frac{1}{R_0}, \quad i_e = \frac{\mu(R_0 - 1)}{\beta} \tag{4.4}$$

4.3.1.2 Modelo Baseado em Indivíduos

O Modelo Baseado em Indivíduos (MBI) é conhecido desde o trabalho de Huston et al. [1988], e consiste de uma estrutura que simula as relações entre indivíduos. Um conjunto de regras define as possíveis interações e como as características individuais podem mudar. Esse modelo permite a análise de um sistema pela observação das propriedades de comportamento adaptativo dos indivíduos em uma população. É possível extrair informação do MBI a respeito da variabilidade entre indivíduos, interações locais, ciclos de vida completos, e mudanças ambientais internas ou externas. Adicionalmente, é fácil representar sistemas físicos ou sociais complexos, embora uma grande quantidade de dados e de computação se façam necessários. Este tipo de modelo pode representar a heterogeneidade das interações sociais, permitindo a análise de redes sociais dinâmicas.

Este trabalho usa o MBI que foi desenvolvido por Almeida [2011], o qual por sua vez inclui melhorias em relação ao modelo proposto por Nepomuceno [2005]. Esta versão foi desenvolvida com o objetivo de reproduzir as premissas do modelo SIR com um modelo estocástico de interações entre indivíduos. Conseqüentemente o MBI converge, em média, para o comportamento do modelo SIR. Algumas premissas epidemiológicas são adotadas aqui. Os parâmetros N , μ , β e γ têm a mesma interpretação e os mesmos valores que no modelo SIR. O número de indivíduos é constante. Cada indivíduo assume, em um instante de tempo, um único estado relacionado com sua característica de saúde. A perda da imunidade, a recuperação e a vacinação são eventos que ocorrem com distribuição uniforme em cada intervalo de tempo discreto. Em cada instante de tempo, cada indivíduo pode mudar de categoria de saúde de acordo com as seguintes regras: (a) $R \rightarrow S$, ou seja, um indivíduo recuperado eventualmente se torna novamente susceptível; (b) $S \rightarrow I$, ou seja, um indivíduo susceptível pode se infectar em decorrência do contato com um indivíduo infectado; (c) $I \rightarrow R$, ou seja, um indivíduo infectado irá se recuperar; (d) $S \rightarrow R$, ou seja, um indivíduo susceptível pode se tornar diretamente recuperado, caso seja vacinado.

4.3.2 Problema de Otimização

Os valores dos parâmetros aqui utilizados simulam uma doença hipotética para a qual existe uma vacina efetiva. Supõe-se que o tempo médio para recuperação seja de 15 u.t., o que significa que $\gamma = 1/15$ (u.t.)⁻¹. O tamanho considerado para a população seria de $N = 1000$ indivíduos. Um indivíduo recuperado iria perder sua imunidade em média após um período de 200 u.t., o que implica $\mu = 1/200$ (u.t.)⁻¹. Considera-se $\beta = 1,5$ (u.t.)⁻¹.

O problema utilizado neste exemplo é uma simplificação daquele considerado em da Cruz et al. [2017], fazendo uso de valores predeterminados para os instantes de aplicação das campanhas de vacinação na fase transitória, que são assumidos iguais a: $\Delta t \in \{0, 7, 7, 7, 14, 14, 14, 14, 14\}$. Depois disso, haverá uma ação periódica a cada 21 u.t., sendo considerado um horizonte de tempo total de $t_f = 720$ u.t.. Um vetor de dez variáveis de decisão $\mathbf{x} = \{x_1, \dots, x_{10}\}$, contém a informação, nas variáveis x_1 a x_9 , das proporções de indivíduos susceptíveis a serem vacinados em cada campanha de vacinação, durante a fase transitória, e a variável x_{10} contém a proporção de vacinação a ser empregada no regime permanente.

Com o objetivo de manter o problema aqui abordado no âmbito da otimização mono-objetivo, a função-objetivo $c(\mathbf{x})$ a ser minimizada, conforme Equação 4.5, é agora formulada como uma função afim do número de infectados ao longo do tempo de análise $i(\mathbf{x})$ e do número de vacinas aplicadas $v(\mathbf{x})$.

$$c(\mathbf{x}) = 1,5i(\mathbf{x}) + v(\mathbf{x}) \tag{4.5}$$

O problema de otimização aqui tratado é então representado pela Equação (4.6).

$$\begin{cases} \min_{\mathbf{x}} c(\mathbf{x}) \\ \text{sujeito a:} \\ \left[\begin{array}{l} [S_t, I_t, R_t] = MBI(S_0, I_0, R_0, t_0, t_f, \mathbf{x}, N_o, \beta_o, \gamma_o) \quad \forall t \in \{t_0, \dots, t_f\} \\ i(\mathbf{x}) = \sum_{t_0}^{t_f} I(t)dt \\ v(\mathbf{x}) = \sum_{t=0}^{t_f} \sum_{i=1}^{10} x_i(t) \\ 0,05 \leq x_i(t) \leq 0,95 \quad \forall t \in \{t_0, \dots, t_f\} \end{array} \right. \end{cases} \tag{4.6}$$

Neste modelo, a função $MBI(S_0, I_0, R_0, t_0, t_f, \mathbf{x}, N_o, \beta_o, \gamma_o)$ retorna os vetores S_t , I_t e R_t , que respectivamente representam os números de susceptíveis, infectados em recuperados em cada instante de tempo. A variável tempo $t \in \{t_0, \dots, t_f\}$ é uma variável discreta, com intervalo de discretização igual a 1 u.t., e as variáveis de decisão $x_i(t)$ são iguais a zero em instantes de tempo distintos daqueles em que estão programadas campanhas de vacinação.

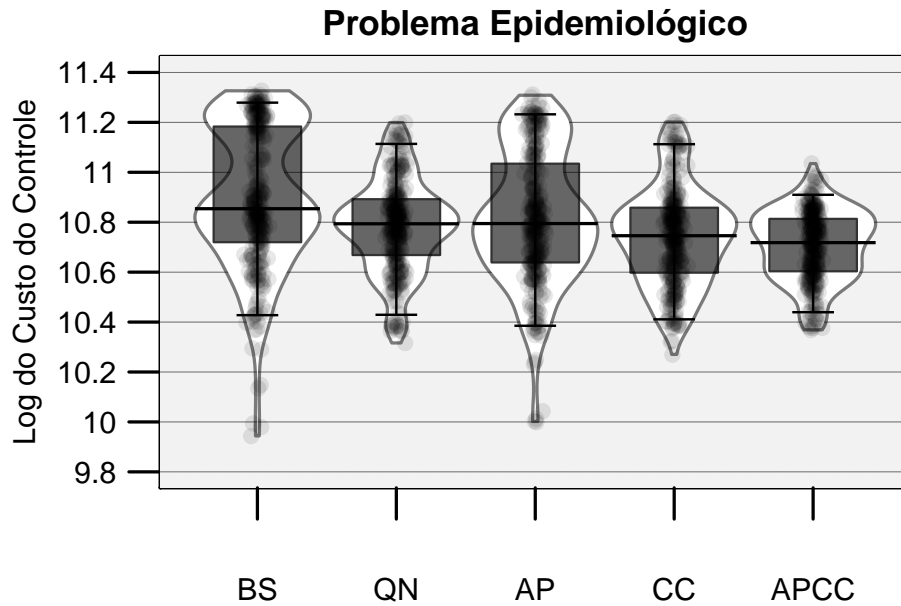


Figura 4.6: Beanplot para o $\log(f(x) + 1)$ do custo das melhores políticas de vacinação retornados pelos algoritmos.

4.3.3 Comparação de Algoritmos

Os algoritmos ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC são agora comparados sobre o problema de otimização (4.6). Cada algoritmo conta com um orçamento de 1000 avaliações de função-objetivo em cada execução, sendo executado 30 vezes. Desta maneira, são obtidos 30 conjuntos independentes de taxas de vacinação otimizadas para cada estratégia. De modo a se extrair uma amostra considerável para a comparação das heurísticas, cada solução retornada será simulada 10 vezes no Modelo Baseado em Indivíduo com diferentes sementes aleatórias. Essa reamostragem do valor da função-objetivo é realizada para permitir a análise da variabilidade de cada solução encontrada diante da variação das diferentes realizações do processo estocástico subjacente ao problema em questão.

As 300 amostras dos valores de função-objetivo das soluções obtidas com as estratégias ES-BS, ES-QN, ES-AP e ES-APCC são apresentadas na Figura 4.1. Este beanplot sugere, para este problema particular, a superioridade das heurísticas baseadas em comparação alicerçada em confiança estatística, que são o ES-CC e o ES-APCC. Destaca-se, dentre os dois, o método ES-APCC que, apesar de não retornar o menor valor avaliado de função-objetivo, obteve uma amostra mais robusta e com menores valores da função-objetivo ruidosa para a maioria dos quantis representados na figura. No caso, a interação entre a busca local baseada na otimização de aproximações lineares-quadráticas juntamente com o operador de comparação proporcionou tal desempenho.

Tabela 4.10: Estatísticas para o $\log(f(x) + 1)$ dos valores de função-objetivo das melhores políticas de vacinação retornadas pelos algoritmos.

Estatística	BS	QN	AP	CC	APCC
Mínimo	9,9431	10,3160	9,9988	10,2694	10,3679
1º Quartil	10,7201	10,6684	10,6390	10,5979	10,6035
Mediana	10,8541	10,7938	10,7946	10,7458	10,7182
3º Quartil	11,1831	10,8921	11,0346	10,8580	10,8135
Máximo	11,3272	11,1996	11,3117	11,2022	11,0356

Tabela 4.11: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação ao logaritmo do valor de função-objetivo das melhores políticas de vacinação encontradas.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,002135	0,136966	–	–
CC	0,000000	0,322130	0,014181	–
APCC	0,000000	0,014181	0,000006	0,136966
Valor-p do Teste: 0,000000				

A Tabela 4.10 mostra as estatísticas sobre o logaritmo dos valores de função-objetivo retornados em relação ao valor mínimo, primeiro, segundo e terceiro quartis, além do valor máximo.

A Tabela 4.11 mostra o resultado do teste de Quade. Essa tabela indica a existência de diferença significativa entre as medianas de melhores custos do ES-APCC e todas as outras estratégias evolutivas, com exceção do ES-CC. O ES-CC por sua vez não apresentou diferença significativa em relação à ES-QN, que por sua vez não diferiu estatisticamente do ES-AP. Na última posição de eficiência se encontra o ES-BS.

O ranking obtido para o ordenamento das medianas do valor da função-objetivo é mostrado na Tabela 4.12. O primeiro lugar nesse ranking é compartilhado entre a heurística ES-APCC e ES-CC. Os conjuntos de soluções para o problema de controle de epidemia gerados pelas estratégias evolutivas são mostrados nas Tabelas B.5, B.6, B.7, B.8 e B.9 do Apêndice B.

Por fim, cabe estudar o comportamento dos algoritmos em relação ao tempo gasto para encontrar a melhor solução deste problema, dado o orçamento de 1000 avaliações de função-objetivo em cada uma das 30 execuções. São apresentados o beanplot da Figura 4.7, que indica a distribuição do tempo gasto em segundos por cada heurística, a Tabela 4.13, com as estatísticas a respeito do tempo de execução, e a Tabela 4.14, com o resultado do teste de Quade ao comparar as estratégias de forma pareada. As medianas indicam que o ES-APCC corresponde à implementação mais rápida, seguida

Tabela 4.12: Ranking da mediana do valor da função-objetivo para as diferentes heurísticas. Considera-se empate o caso de diferença não-significativa, para significância de 0,05. Por convenção, se ocorre empate de duas ou mais heurísticas em determinado ranking, uma heurística situada logo abaixo recebe ranking um a mais se ficar empatada com uma das anteriores, ou ranking igual ao número de heurísticas anteriores mais um se for significativamente diferente das anteriores.

	ES-BS	ES-QN	ES-AP	ES-CC	ES-APCC
Problema (4.6)	5	2	2	1	1

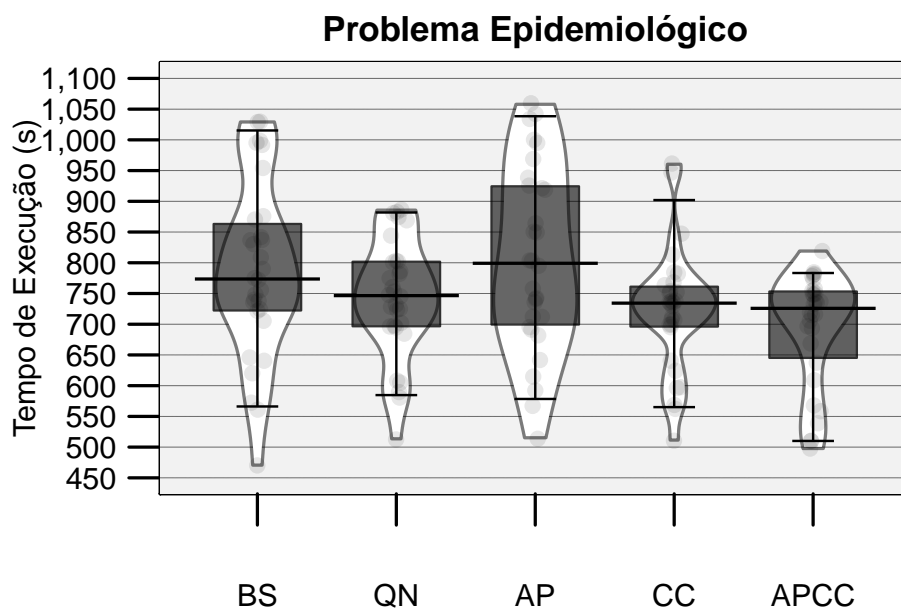


Figura 4.7: Beanplot para o tempo de execução dos algoritmos, em segundos.

Tabela 4.13: Estatísticas para o tempo de execução dos algoritmos, em segundos.

Estatística	BS	QN	AP	CC	APCC
Mínimo	470,2310	512,4290	513,5540	510,3730	497,7220
1º Quartil	722,3967	696,6223	699,5065	696,1682	645,1875
Mediana	773,6085	746,5455	799,1470	734,2930	725,8165
3º Quartil	863,2360	801,7025	924,4945	761,0880	753,2140
Máximo	1029,5520	886,7300	1059,5110	961,4720	819,0880

pelo ES-CC, ES-QN, ES-BS e ES-AP. No entanto, somente foi detectada diferença significativa no tempo de execução entre o ES-APCC e o ES-AP.

Vale comentar que não se verificou um puro e simples aumento sistemático do tempo de execução dos algoritmos relacionado a um possível *overhead* dos procedimentos de aproximação de funções ou de amostragem. Pelo contrário, neste exemplo foi revelado um padrão complexo de interação entre as distintas componentes dos algorit-

Tabela 4.14: Valores-p geral e das comparações pareadas entre os tempos de execução dos algoritmos, em segundos, pelo teste de Quade para determinar as melhores políticas de vacinação.

	BS	QN	AP	CC
QN	0,977771	–	–	–
AP	0,977771	0,977771	–	–
CC	0,543551	0,825250	0,120408	–
APCC	0,043095	0,092245	0,004909	0,977771
Valor-p do Teste: 0,003562				

mos, que parecem levar o algoritmo ES-AP aos maiores tempos na maioria dos quantis, mas também sugerem que o algoritmo ES-APCC apresente os menores tempos para a maioria dos quantis.

De maneira geral, o resultado obtido deste terceiro experimento guarda consistência com aqueles do segundo experimento, que indicava a superioridade dos algoritmos ES-AP e ES-APCC no contexto da otimização de funções-objetivo ruidosas, sugerindo a conveniência da adoção de um algoritmo dotado de busca local por aproximação quadrática. Neste terceiro experimento, aparece uma indicação de superioridade dos algoritmos ES-APCC e ES-CC no contexto da otimização de um problema ruidoso específico de otimização da política de vacinação para controle de uma epidemia. Essa indicação sugere a conveniência da utilização de um mecanismo de reamostragem dos valores da função-objetivo ruidosa, visando o aumento da confiança na tomada de decisão dentro do algoritmo de busca. Em comum aos dois experimentos, parece emergir a indicação do uso da heurística ES-APCC como primeira alternativa para abordar problemas ruidosos, na falta de conhecimento mais específico sobre o comportamento dos diferentes algoritmos no problema a ser tratado. Soma-se a isso o aspecto do tempo computacional requerido para a convergência, que também reforça a indicação da heurística ES-APCC. Deve-se pontuar, entretanto, que a confirmação de tal indicação ainda requer a realização de testes mais exaustivos, utilizando classes mais diversificadas de problemas ruidosos, o que é deixado aqui como uma proposta de trabalho a ser realizado no futuro.

Capítulo 5

Conclusões

Esta tese enfocou, como tema central de investigação, a questão da otimização de funções ruidosas por algoritmos evolutivos. A otimização de tal classe de funções por meio de algoritmos evolutivos tem sido abordada já há alguns anos. Alguns trabalhos procuraram examinar a capacidade intrínseca desses algoritmos para tratar funções dessa classe. A maioria dos trabalhos mais recentes, no entanto, procura estudar a aplicação de técnicas de reamostragem do valor de função-objetivo, visando aumentar a confiança nos processos de tomada de decisão internos do algoritmo, desta forma mitigando o efeito do ruído.

Este trabalho de tese delimitou, como hipótese a ser investigada, a possibilidade de que a utilização de funções de aproximação na construção de mecanismos de busca local pudesse também ter um papel de reduzir os efeitos do ruído, em virtude de uma filtragem que seria realizada implicitamente pelo processo de construção da função de aproximação.

Para o estudo da questão proposta para investigação, foram considerados os seguintes algoritmos:

- Foi escolhida uma classe de algoritmos evolutivos bastante simples, a classe das Estratégias Evolutivas, para servir de base para experimentos numéricos. Tal escolha foi feita procurando evitar que interações complexas entre diferentes componentes de um algoritmo mais elaborado dificultassem a análise do efeito dos mecanismos a serem estudados. O algoritmo básico pertencente a essa classe foi denominado ES-BS.
- A esse algoritmo ES-BS foi acrescido um mecanismo clássico de busca local baseado em direção de busca, conduzindo ao algoritmo denominado ES-QN.

- Um outro algoritmo foi formulado como modificação do ES-BS, agora com o acréscimo de um mecanismo de reamostragem da função-objetivo, de maneira a constituir um algoritmo representativo da abordagem atualmente predominante no tratamento de funções ruidosas por algoritmos evolutivos. Esse algoritmo foi denominado ES-CC.
- Um algoritmo foi então formulado também a partir do ES-BS de forma a incluir um mecanismo de busca local baseada em aproximações quadráticas e lineares. Esse algoritmo, denominado ES-AP, permitiria a análise do efeito da inclusão do mecanismo proposto nesta tese dentre os operadores do algoritmo evolutivo básico.
- Por fim, foi formulado um algoritmo que, além de incluir o operador de busca local baseada em aproximações de função, também incluía o operador de reamostragem, assim combinando a nova estratégia aqui proposta com a estratégia hoje predominante no tratamento de funções ruidosas. Esse algoritmo foi denominado ES-APCC.

Esse conjunto de cinco algoritmos permitiu estabelecer uma série de bases de comparação para fundamentar o estudo aqui proposto. Os algoritmos ES-BS, ES-QN e ES-CC serviriam de base de comparação representativas das principais alternativas hoje disponíveis: um algoritmo evolutivo básico, um outro evolutivo dotado de busca local baseada no algoritmo Quasi-Newton e um algoritmo evolutivo dotado de técnica de reamostragem. O algoritmo ES-AP constituiria, por sua vez, o protótipo da pura aplicação do mecanismo de busca aqui proposto como operador evolutivo adicional. Por fim, o algoritmo ES-APCC permitiria a análise da interação das duas formas para tratamento de ruído: o procedimento aqui proposto, baseado na filtragem por meio de uma função de aproximação, e a abordagem hoje predominante de tratamento de ruído, baseada em reamostragem do valor de função-objetivo.

Um problema metodológico importante se apresentou, precedendo a realização de experimentos numéricos para comparar os algoritmos assim construídos. O problema era: como assegurar uma comparação justa entre os diferentes algoritmos, uma vez que cada algoritmo possui um conjunto de parâmetros, de cujo ajuste depende o seu desempenho? Deve-se notar que não seria possível, no contexto do problema aqui delimitado, recorrer a ajustes padronizados de parâmetros para os algoritmos que serviriam de base de comparação, uma vez que: (i) dois dos algoritmos a serem utilizados como base de comparação, o ES-QN e o ES-CC, não seriam exatamente análogos a outros algoritmos estudados na literatura, o que impediria o reaproveitamento de ajustes de parâmetros

obtidos anteriormente; (ii) mesmo o algoritmo clássico ES-BS não dispõe de ajustes de parâmetros padronizados para o caso de funções ruidosas; e (iii) mesmo que existissem ajustes previamente conhecidos para os parâmetros dos três algoritmos que serviriam de base de comparação, ainda assim persistiria a objeção quanto à possibilidade de que os algoritmos propostos, ES-AP e ES-APCC, tivessem seus parâmetros ajustados de maneira a especificamente favorecer a otimização do conjunto particular de funções de teste a ser empregado neste estudo.

Por todos esses motivos, foi desenvolvido nesta tese um mecanismo automático de ajuste de parâmetros para algoritmos estocásticos de otimização. Para sanar as dificuldades apontadas acima, bastaria: (i) que tal mecanismo de ajuste fosse “razoável”, no sentido de que pudesse extrair um bom desempenho dos algoritmos cujos parâmetros fossem sintonizados, ou seja, evitando deixar que os algoritmos fossem sintonizados para operar com desempenho muito abaixo de seu potencial, o que prejudicaria a comparação entre algoritmos pelo fato de que cada algoritmo poderia estar a uma distância diferente de seu melhor desempenho possível; e (ii) que tal mecanismo fosse empregado exatamente nas mesmas condições para todos os algoritmos a serem comparados. No entanto, para além desses requisitos mínimos necessários para o estabelecimento de condições para a comparação de algoritmos, o mecanismo de ajuste aqui proposto foi comparado com diversos mecanismos de ajuste hoje disponíveis na literatura, tendo se mostrado superior a todos estes. Espera-se, portanto, que os algoritmos a serem analisados neste trabalho de tese tenham sido sintonizados de maneira a apresentarem desempenho próximo àquele que podem idealmente atingir.

Com os algoritmos devidamente sintonizados, três experimentos numéricos foram conduzidos. Primeiro, foram realizados testes sobre funções de *benchmark* que vêm sendo utilizadas como referência para o estudo de algoritmos evolutivos. Essas funções são dotadas de características distintas que incluem: bom-condicionamento ou mal-condicionamento, convexidade ou não-convexidade, unimodalidade ou multimodalidade, estrutura deceptiva ou dotada de tendência global, e outras. Todas as funções, entretanto, são isentas de ruído. Neste conjunto de testes, portanto, não poderiam ser avaliados os algoritmos ES-CC e ES-APCC, os quais incluem uma etapa de reamostragem de soluções. A comparação se restringia aos algoritmos ES-BS, ES-QN e ES-AP, e tinha por objetivo avaliar a influência do operador proposto de busca local por aproximação de funções nas propriedades de convergência do algoritmo em circunstâncias nas quais os fatores determinantes do comportamento de algoritmos evolutivos são relativamente bem conhecidos.

Seria de se esperar que, nesta etapa, o algoritmo ES-AP apresentasse comportamento essencialmente equivalente ao ES-QN e um pouco superior ao ES-BS, o que

estaria de acordo com o padrão usualmente verificado que indica que a utilização de buscas locais tende a melhorar o desempenho de algoritmos evolutivos. No entanto, os resultados alcançados nos testes sugerem que o algoritmo ES-AP tenha obtido desempenho superior ao ES-QN – ressaltando-se a necessidade de testes mais exaustivos para dar suporte a uma avaliação mais conclusiva. Conjectura-se que a formulação explícita de funções de aproximação que representem o conjunto de amostras já avaliadas da função-objetivo, conforme feito no ES-AP, permita codificar a informação a respeito de tendências de longa distância presentes nas funções de uma maneira mais útil que a codificação implícita presente na formulação Quasi-Newton do ES-QN, sendo ambas por sua vez melhores que a ausência de mecanismos de modelagem de função do ES-BS. Cabe projetar testes mais específicos que permitam examinar esta ou outras conjecturas.

A seguir, foram realizados testes sobre as mesmas funções de *benchmark* empregadas na etapa anterior, agora transformadas em funções ruidosas pelo acréscimo de ruído Gaussiano aditivo no momento de sua avaliação. Desta vez, todos os algoritmos considerados nesta tese são avaliados: ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC. Os novos algoritmos agora acrescentados, ES-CC e ES-APCC, contam com um operador de reamostragem de funções que é utilizado no contexto das tomadas de decisão no interior do algoritmo, sendo nos demais aspectos semelhantes respectivamente ao ES-BS e ao ES-AP.

Seria de se esperar que, nessas condições, o algoritmo ES-CC tivesse desempenho melhor que o ES-BS. Isso não aconteceu, sendo que os resultados não foram conclusivos no que diz respeito à comparação entre esses algoritmos. Também seria de se esperar que o algoritmo ES-APCC tivesse desempenho superior ao ES-AP. Também esse resultado não foi obtido, tendo sido verificado precisamente o contrário: o ES-AP foi superior ao ES-APCC nesse conjunto de testes. Isso sugere que ou existe redundância entre os efeitos da reamostragem e da filtragem implícita por função de aproximação, o que poderia significar que existiria um desperdício do orçamento de avaliações de função disponível no caso da realização das duas operações, ou então que o operador de reamostragem não produz efeito relevante que compense o dispêndio de avaliações em sua execução. Seria de se esperar ainda que o algoritmo ES-QN tivesse seu desempenho mais degradado nesse novo cenário, em comparação com o cenário do experimento anterior, do que os algoritmos ES-BS e ES-AP, dada a reconhecida fragilidade da operação de cálculo de gradiente diante da presença de ruído. Essa previsão se confirmou, com o algoritmo ES-QN saindo de uma posição de aparente melhor desempenho em relação ao ES-BS no experimento anterior para uma posição de desempenho aparentemente pior que este no segundo experimento. O algoritmo ES-AP, por outro lado,

passou a se destacar mais claramente dos outros dois, no novo cenário.

A principal pergunta a ser respondida por este experimento dizia respeito à questão de qual procedimento seria mais efetivo para o tratamento de ruídos, se a reamostragem de soluções ou a filtragem implícita por uma aproximação de funções, expressa na comparação entre o ES-CC e o ES-AP. No contexto do conjunto de funções que foi considerado, o resultado foi claramente favorável ao algoritmo ES-AP, o que daria suporte à principal hipótese considerada neste trabalho de tese: tais evidências sugeririam que a filtragem implícita por meio de funções de aproximação seria mais efetiva que o procedimento de reamostragem. Outra questão conexa dizia respeito à possível interação entre os operadores de reamostragem e de aproximação de funções, que estava implementada no algoritmo ES-APCC. Embora, como comentado anteriormente, a análise do conjunto dos dados sugira que na maioria dos casos bastaria o operador de aproximação de funções, como no algoritmo ES-AP, um detalhe nos resultados chama a atenção: nas poucas situações em que o algoritmo ES-AP não ficou posicionado em primeiro lugar nos testes, o algoritmo que assumiu a primeira posição foi o ES-APCC. Isso sugere que em determinadas situações a interação entre os dois efeitos possa ser vantajosa.

Por fim, foi realizado um conjunto de testes sobre um problema com motivação prática – o projeto de campanhas de vacinação para o controle de epidemias. Esse problema, conforme formulado nesta tese, trata-se de um problema que tem estrutura fundamentalmente ruidosa, sendo um representante real da classe de problemas para a qual os métodos aqui desenvolvidos foram concebidos. Novamente foram analisados os desempenhos de todos os algoritmos aqui considerados, ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC. Neste cenário, os resultados obtidos sugerem conclusões um pouco diferentes daquelas extraídas do segundo experimento. Desta vez, o algoritmo ES-APCC exibiu o melhor desempenho entre todos, superando o ES-AP. Esse resultado não necessariamente conflita com os resultados do experimento 2, uma vez que também nesse experimento ocorreram casos de funções para as quais o algoritmo de melhor desempenho foi o ES-APCC. Um padrão diferente do observado no experimento 2, no entanto, diz respeito ao algoritmo ES-CC, que no experimento 3 não exibiu desempenho significativamente pior que o algoritmo ES-APCC, situando-se também em primeira colocação. No experimento 2, não havia sido observada nenhuma situação em que tal algoritmo superava o ES-AP. Esses resultados sugerem que a operação de reamostragem possa desempenhar papel de destacada relevância em determinadas circunstâncias, em problemas de otimização de funções ruidosas. Deve-se notar que essa observação ocorreu precisamente no caso de um problema que é estruturalmente nativo do contexto ruidoso, diferentemente da observação que contraria essa conclusão, extraída de um

contexto de funções que são tornadas ruidosas de maneira artificial.

Não obstante algumas indicações contraditórias extraídas dos diversos experimentos, uma análise do conjunto dos resultados obtidos nesta tese sugere que o operador de aproximação linear-quadrática de funções possa constituir importante componente para a montagem de algoritmos evolutivos orientados para o tratamento de funções ruidosas. Nos três experimentos realizados, o algoritmo posicionado em primeiro lugar sempre fez uso desse operador. Faltam certamente análises mais detalhadas das circunstâncias que tornam necessária ou dispensável a utilização de um operador de reamostragem.

5.1 Trabalhos Futuros

A própria discussão conduzida na seção anterior indica um conjunto de questões que foram deixadas em aberto por este trabalho de tese. Os seguintes estudos parecem prioritários:

- Seria interessante investigar a construção de problemas *benchmark* para problemas ruidosos, de tal forma que características existentes em problemas reais do tipo ruidoso pudessem ser artificialmente introduzidas nos problemas *benchmark*, de maneira controlada.
- Na mesma linha, seria importante investigar quais características de problemas ruidosos interferem de maneira mais significativa nas propriedades de convergência de algoritmos empregados para sua otimização. Essa investigação, em princípio, ocorreria junto com a pesquisa a respeito da construção de problemas *benchmark*, com a colocação dessa questão que é complementar à anterior.
- Completando o conjunto de questões a respeito do ambiente de testes que se faz necessário para viabilizar a pesquisa sobre otimização de funções ruidosas, seria ainda importante identificar, catalogar, estudar e produzir implementações padrão de uma variedade de problemas ruidosos reais com características distintas, de tal forma a possibilitar procedimentos de validação de novos algoritmos para otimização desta classe de funções.
- Sabe-se que seria possível realizar procedimentos de reamostragem de maneiras diversas. Da mesma forma, seria possível construir operadores de aproximação de função que constituíssem diferentes variações de um procedimento geral. Seria importante produzir análises que considerassem uma maior variedade de operadores de reamostragem, bem como uma maior variedade de operadores de busca

local por aproximação de função, de maneira a permitir um panorama mais detalhado de análises sobre os mecanismos subjacentes aos efeitos observados.

- Na presente tese, o mecanismo de busca local por aproximação de função foi justaposto ao mecanismo de reamostragem, para a construção do algoritmo ES-APCC. Seria importante investigar a possibilidade de articulação dos dois mecanismos segundo processos mais complexos que a sua simples justaposição, explorando possibilidades de uma efetiva hibridização de operadores. Um exemplo de possibilidade de tal hibridização seria a possível utilização da informação proveniente da função aproximada (que corresponde a uma filtragem implícita da função) para a tomada de decisão no interior do operador de reamostragem.
- A seguir, deve-se mencionar a necessidade de estudos que avaliem a transposição das técnicas aqui propostas para algoritmos evolutivos de tipos diferentes das Estratégias Evolutivas. Seria importante a realização de testes envolvendo as principais famílias de algoritmos evolutivos para problemas contínuos, tais como os algoritmos de Evolução Diferencial, os Algoritmos Imunológicos, as diferentes modalidades de Algoritmos Genéticos, algoritmos do tipo GRASP (Greedy Randomized Adaptive Search Procedure) e ILS (Iterated Local Search), dentre outros.
- Por fim, como questão prospectiva, deve-se mencionar a necessidade de se investigar a possibilidade de transposição do procedimento de aproximação de funções para o contexto de otimização combinatória.

Referências Bibliográficas

- Aizawa, A. N. & Wah, B. W. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2):97--122.
- Akimoto, Y.; Astete-Morales, S. & Teytaud, O. (2015). Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. *Theoretical Computer Science*, 605:42--50.
- Almeida, A. C. L. (2011). *Modelos matemáticos de propagação de epidemias baseados em redes sociais e detecção de clusters de doenças*. Tese de doutorado, Programa de Pós-Graduação em Engenharia Elétrica.
- Anderson, R. M. & May, R. M. (1992). *Infectious diseases of humans: dynamics and control*. Oxford University Press.
- Araujo, M. C.; Wanner, E. F.; Guimarães, F. G. & Takahashi, R. H. (2009). Constrained optimization based on quadratic approximations in genetic algorithms. Em *Constraint-handling in evolutionary optimization*, pp. 193--217. Springer.
- Arnold, D. V. & Beyer, H.-G. (2003a). On the benefits of populations for noisy optimization. *Evolutionary Computation*, 11(2):111--127.
- Arnold, D. V. & Beyer, H.-G. (2003b). On the effects of outliers on evolutionary optimization. Em *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 151--160. Springer.
- Asouti, V.; Kampolis, I. C. & Giannakoglou, K. C. (2009). A grid-enabled asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization. *Genetic Programming and Evolvable Machines*, 10(4):373.
- Auger, A. & Hansen, N. (2005a). Performance evaluation of an advanced local search evolutionary algorithm. Em *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pp. 1777--1784. IEEE.

- Auger, A. & Hansen, N. (2005b). A restart CMA evolution strategy with increasing population size. Em *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pp. 1769--1776. IEEE.
- Auger, A.; Schoenauer, M. & Vanhaecke, N. (2004). LS-CMA-ES: A second-order algorithm for covariance matrix adaptation. Em *International Conference on Parallel Problem Solving from Nature*, pp. 182--191. Springer.
- Bäck, T.; Foussette, C. & Krause, P. (2013). *Contemporary evolution strategies*. Springer.
- Bandler, J. W.; Biernacki, R.; Chen, S.; Song, J.; Ye, S. & Zhang, Q. (1991). Gradient quadratic approximation scheme for yield-driven design. Em *Microwave Symposium Digest, 1991., IEEE MTT-S International*, pp. 1197--1200. IEEE.
- Bayer, P.; de Paly, M. & Bürger, C. M. (2010). Optimization of high-reliability-based hydrological design problems by robust automatic sampling of critical model realizations. *Water Resources Research*, 46(5).
- Bayram, S. & Gezici, S. (2009). Noise-enhanced m-ary hypothesis-testing in the minimax framework. Em *Signal Processing and Communication Systems, 2009. ICSPCS 2009. 3rd International Conference on*, pp. 1--6. IEEE.
- Benigno, P. & Woodford, M. (2012). Linear-quadratic approximation of optimal policy problems. *Journal of Economic Theory*, 147(1):1--42.
- Bertsekas, D. P. (2016). *Nonlinear programming*. Athena Scientific Belmont, 3 edição.
- Beyer, H.-G. (2000). Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2):239--267.
- Beyer, H.-G. & Schwefel, H.-P. (2002). Evolution strategies—a comprehensive introduction. *Natural Computing*, 1(1):3--52.
- Bhattacharyya, P. & Willment, T. (1988). Quadratic approximations in the displacement discontinuity method. *Engineering Analysis*, 5(1):28--35.
- Biernacki, R.; Bandler, J.; Song, J. & Zhang, Q.-J. (1989). Efficient quadratic approximation for statistical design. *Circuits and Systems, IEEE Transactions on*, 36(11):1449--1454.

- Biles, J. A. (1994). Genjam: A genetic algorithm for generating jazz solos. Em *ICMC*, volume 94, pp. 131--137.
- Bioucas-Dias, J. M. & Figueiredo, M. A. (2010). Multiplicative noise removal using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(7):1720--1730.
- Birattari, M.; Yuan, Z.; Balaprakash, P. & Stützle, T. (2010). F-race and iterated F-race: An overview. Em *Experimental methods for the analysis of optimization algorithms*, pp. 311--336. Springer.
- Björck, Å. (1996). *Numerical methods for least squares problems*. SIAM.
- Bochkanov, S. (2016). Alglib v3. <http://www.alglib.net>.
- Boubekur, T. & Schlick, C. (2007). QAS: Real-time quadratic approximation of subdivision surfaces. Em *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pp. 453--456. IEEE.
- Boyd, S. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Bull, L. (1999). On model-based evolutionary computation. *Soft Computing*, 3(2):76--82.
- Cao, J. & Voth, G. A. (1995). Modeling physical systems by effective harmonic oscillators: The optimized quadratic approximation. *The Journal of Chemical Physics*, 102:3337.
- Chang, K.-H. (2012). Stochastic Nelder–Mead simplex method—a new globally convergent direct search method for simulation optimization. *European Journal of Operational Research*, 220(3):684--694.
- Chen, X. M.; Zhang, L.; He, X.; Xiong, C. & Li, Z. (2014). Surrogate-based optimization of expensive-to-evaluate objective for optimal highway toll charges in transportation network. *Computer-Aided Civil and Infrastructure Engineering*, 29(5):359--381.
- Cho, S.-B. (2004). Emotional image and musical information retrieval with interactive genetic algorithm. *Proceedings of the IEEE*, 92(4):702--711.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. Wiley, 3 edição. ISBN 978-0471160687.

- Custódio, F. L.; Barbosa, H. J. & Dardenne, L. E. (2010). Full-atom ab initio protein structure prediction with a genetic algorithm using a similarity-based surrogate model. Em *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1--8. IEEE.
- Cutello, V.; Narzisi, G. & Nicosia, G. (2006). A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of The Royal Society Interface*, 3(6):139--151.
- da Cruz, A. R.; Cardoso, R. T. N. & Takahashi, R. H. C. (2011a). Multiobjective dynamic optimization of vaccination campaigns using convex quadratic approximation local search. Em *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 404--417. Springer.
- da Cruz, A. R.; Cardoso, R. T. N. & Takahashi, R. H. C. (2017). Multiobjective synthesis of robust vaccination policies. *Applied Soft Computing*, 50(1):34--47.
- da Cruz, A. R.; Wanner, E. F.; Cardoso, R. T. N. & Takahashi, R. H. C. (2011b). Using convex quadratic approximation as a local search operator in evolutionary multiobjective algorithms. Em *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 1217--1224. IEEE.
- Das, S.; Konar, A. & Chakraborty, U. K. (2005). Improved differential evolution algorithms for handling noisy optimization problems. Em *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pp. 1691--1698. IEEE.
- Das, S. & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4--31.
- De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Tese de doutorado, University of Michigan Ann Arbor, MI, USA, Ann Arbor, MI, USA.
- Den Hertog, D.; De Klerk, E. & Roos, K. (2002). On convex quadratic approximation. *Statistica Neerlandica*, 56(3):376--385.
- Eguchi, S.; Matsugano, Y.; Sakaguchi, H.; Ono, S.; Fukuda, H.; Furukawa, R. & Kawasaki, H. (2015). A comparative study on self-adaptive differential evolution algorithms for test functions and a real-world problem. Em *International Conference on Learning and Intelligent Optimization*, pp. 131--136. Springer.

- Eiben, A. E. & Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19--31.
- Elsayed, S.; Hamza, N. & Sarker, R. (2016). Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems. Em *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pp. 2966--2973. IEEE.
- Emiris, I. Z. & Nicolopoulos, D. (2013). Randomized kd-trees for approximate nearest neighbor search. *CGL-TR-78, NKUA, Tech. Rep.*
- Févotte, C.; Torrèsani, B.; Daudet, L. & Godsill, S. J. (2008). Sparse linear regression with structured priors and application to denoising of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):174--185.
- Fitzpatrick, J. M. & Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine Learning*, 3(2-3):101--120.
- Fonseca, C. H. & Wanner, E. F. (2016). A quadratic approximation-based local search operator for handling two equality constraints in continuous optimization problems. Em *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pp. 4911--4917. IEEE.
- Friedman, J. H.; Bentley, J. L. & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209--226.
- Friedrich, T.; Kötzing, T.; Krejca, M. & Sutton, A. M. (2015). The benefit of sex in noisy evolutionary search. *arXiv preprint arXiv:1502.02793*.
- Gao, Y. & Wang, X. (2009). Surrogate-based process optimization for reducing warpage in injection molding. *Journal of Materials Processing Technology*, 209(3):1302--1309.
- García, S.; Fernández, A.; Luengo, J. & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044--2064.
- Glasmachers, T.; Schaul, T.; Yi, S.; Wierstra, D. & Schmidhuber, J. (2010). Exponential natural evolution strategies. Em *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 393--400. ACM.
- Gough, B. (2009). *GNU scientific library reference manual*. Network Theory Ltd.

- Granichin, O. (2004). Linear regression and filtering under nonstandard assumptions (arbitrary noise). *IEEE Transactions on Automatic Control*, 49(10):1830--1837.
- Grefenstette, J. & Fitzpatric, J. (1985). Genetic search with approximate fitness evaluations. Em *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pp. 112--120.
- Grierson, D. & Pak, W. (1993). Optimal sizing, geometrical and topological design using a genetic algorithm. *Structural Optimization*, 6(3):151--159.
- Grimm, V. & Railsback, S. F. (2005). *Individual-based modeling and ecology*. Princeton University Press.
- Groenwold, A. A. & Etman, L. (2010). A quadratic approximation for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 82(4):505--524.
- Guennebaud, G.; Jacob, B. et al. (2010). Eigen v3. <http://eigen.tuxfamily.org>.
- Guo, S.-M.; Tsai, J. S.-H.; Yang, C.-C. & Hsu, P.-H. (2015). A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. Em *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pp. 1003--1010. IEEE.
- Gutiérrez-Antonio, C. & Briones-Ramírez, A. (2016). Multiobjective stochastic optimization of dividing-wall distillation columns using a surrogate model based on neural networks. *Chemical and Biochemical Engineering Quarterly*, 29(4):491--504.
- Hansen, N.; Arnold, D. V. & Auger, A. (2015). Evolution strategies. Em *Springer Handbook of Computational Intelligence*, pp. 871--898. Springer.
- Hansen, N. & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. Em *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 312--317. IEEE.
- Hansen, N. & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159--195.
- Hansen, N.; Ostermeier, A. & Gawelczyk, A. (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. Em *ICGA*, pp. 57--64.

- Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, 42(4):599--653.
- Hinman, A. (1999). Eradication of vaccine-preventable diseases. *Annual Review of Public Health*, 20(1):211--229.
- Horn, S.-C. & Lin, S.-Y. (2013). Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation. *Information Sciences*, 233:214--229.
- Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339--342.
- Hu, L.; Nooshabadi, S. & Ahmadi, M. (2016). Parallel randomized KD-tree forest on GPU cluster for image descriptor matching. Em *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pp. 582--585. IEEE.
- Hussain, M. S.; Javadi, A. A.; Ahangar-Asr, A. & Farmani, R. (2015). A surrogate model for simulation-optimization of aquifer systems subjected to seawater intrusion. *Journal of Hydrology*, 523:542--554.
- Huston, M.; DeAngelis, D. & Post, W. (1988). New computer models unify ecological theory. *BioScience*, 38(10):682--691.
- Hutama, S. A.; Nugroho, S. & Utomo, D. (2016). Features deletion on multiple objects recognition using speeded-up robust features, scale invariant feature transform and randomized KD-tree. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(2):692--698.
- Hutter, F.; Hoos, H. H.; Leyton-Brown, K. & Stützle, T. (2009). ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267--306.
- Hutter, F.; Hoos, H. H. & Stützle, T. (2007). Automatic algorithm configuration based on local search. Em *AAAI*, volume 7, pp. 1152--1157.
- Hüsken, M.; Jin, Y. & Sendhof, B. (2005). Structure optimization of neural networks for evolutionary design optimization. *Soft Computing*, 9(1):21--28.
- Igel, C.; Suttrop, T. & Hansen, N. (2006). A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. Em *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 453--460. ACM.

- Jakobi, N.; Husbands, P. & Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. Em *European Conference on Artificial Life*, pp. 704--720. Springer.
- Jamai, R. & Damil, N. (2001). Quadratic approximants in the asymptotic numerical method. *Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics*, 329(11):809--814.
- Jastrebski, G. A. & Arnold, D. V. (2006). Improving evolution strategies through active covariance matrix adaptation. Em *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 2814--2821. IEEE.
- Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61--70.
- Jin, Y. & Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3):303--317.
- Jin, Y.; Olhofer, M. & Sendhoff, B. (2000). On evolutionary optimization with approximate fitness functions. Em *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pp. 786--793. Morgan Kaufmann Publishers Inc.
- Jin, Y.; Olhofer, M. & Sendhoff, B. (2001). Managing approximate models in evolutionary aerodynamic design optimization. Em *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pp. 592--599. IEEE.
- Johanson, B. & Poli, R. (1998). *GP-music: An interactive genetic programming system for music generation with automated fitness raters*. University of Birmingham, Cognitive Science Research Centre.
- Jun-hua, L. & Ming, L. (2013). An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments. *Optik-International Journal for Light and Electron Optics*, 124(24):6780--6785.
- Kampstra, P. et al. (2008). Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software*, 28(1):1--9.
- Kermack, W. O. & McKendrick, A. G. (1933). Contributions to the mathematical theory of epidemics. iii. further studies of the problem of endemicity. *Proceedings of the Royal Society of London. Series A*, 141(843):94--122.

- Kim-Farley, R. (1992). Global immunization. *Annual Review of Public Health*, 13(1):223--237.
- Kimura, T. (2010). Valuing executive stock options: A quadratic approximation. *European Journal of Operational Research*, 207(3):1368--1379.
- Kiwiel, K. C. (1984). A quadratic approximation method for minimizing a class of quasidifferentiable functions. *Numerische Mathematik*, 45(3):411--430.
- Koziel, S. & Bekasiewicz, A. (2016). Multiobjective antenna design by means of sequential domain patching. *IEEE Antennas and Wireless Propagation Letters*, 15:1089--1092.
- Krämer, A.; Hülsmann, M.; Köddermann, T. & Reith, D. (2014). Automated parameterization of intermolecular pair potentials using global optimization techniques. *Computer Physics Communications*, 185(12):3228--3239.
- Krink, T.; Filipic, B. & Fogel, G. B. (2004). Noisy optimization problems-a particular challenge for differential evolution? Em *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pp. 332--339. IEEE.
- Lee, J. & Hajela, P. (1996). Parallel genetic algorithm implementation in multidisciplinary rotor blade design. *Journal of Aircraft*, 33(5):962--969.
- Li, C.; Wang, F.-L.; Chang, Y.-Q. & Liu, Y. (2010). A modified global optimization method based on surrogate model and its application in packing profile optimization of injection molding process. *The International Journal of Advanced Manufacturing Technology*, 48(5-8):505--511.
- Li, J.; Liu, P.; Zhou, L. & Li, M. (2014). An improvement to genetic algorithms for multimodal optimization in noisy environments: Re-evaluation of all individuals per generation. Em *Computational Intelligence and Communication Networks (CICN), 2014 International Conference on*, pp. 657--661. IEEE.
- Liang, J.; Qu, B. & Suganthan, P. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*.

- Liao, T. & Stuetzle, T. (2013). Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. Em *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 1938--1944. IEEE.
- Liao, X.; Li, Q.; Yang, X.; Zhang, W. & Li, W. (2008). Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35(6):561--569.
- Logvinenko, A. D. (2003). Method of quadratic approximation: A new approach to identification of analysers and channels in human vision. *Journal of Mathematical Psychology*, 47(5):495--506.
- Loshchilov, I. (2013). CMA-ES with restarts for solving CEC 2013 benchmark problems. Em *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 369--376. Ieee.
- Ma, J.; Zhang, J. & Zhou, M. (2015). Group decision-making inspired particle swarm optimization in noisy environment. Em *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pp. 316--321. IEEE.
- Marcia, R. F.; Mitchell, J. C. & Rosen, J. B. (2005). Iterative convex quadratic approximation for global optimization in protein docking. *Computational Optimization and Applications*, 32(3):285--297.
- Marzat, J.; Walter, E. & Piet-Lahanier, H. (2013). Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, 55(4):707.
- Matloff, N. (2011). *The art of R programming: A tour of statistical software design*. No Starch Press.
- Mendes, M. H.; Soares, G. L.; Coulomb, J.-L. & Vasconcelos, J. A. (2013). A surrogate genetic programming based model to facilitate robust multi-objective optimization: A case study in magnetostatics. *IEEE Transactions on Magnetics*, 49(5):2065--2068.
- Merelo, J.; Chelly, Z.; Mora, A.; Fernández-Ares, A.; Esparcia-Alcázar, A. I.; Cotta, C.; de las Cuevas, P. & Rico, N. (2016). A statistical approach to dealing with noisy fitness in evolutionary algorithms. Em *Computational Intelligence*, pp. 79--95. Springer.
- Merelo, J. J.; Liberatore, F.; Ares, A. F.; García, R.; Chelly, Z.; Cotta, C.; Rico, N.; Mora, A. M. & García-Sánchez, P. (2015). There is noisy lunch: A study of noise in

- evolutionary optimization problems. Em *Computational Intelligence (IJCCI), 2015 7th International Joint Conference on*, volume 1, pp. 261--268. IEEE.
- Merelo, J. J.; Valdivieso, P. A. C.; Mora, A. M.; Fernández-Ares, A.; Esparcia-Alcázar, A.; Cotta, C. & Rico, N. (2014). Studying and tackling noisy fitness in evolutionary design of game characters. Em *IJCCI (ECTA)*, pp. 76--85.
- Miller, B. L. (1997). *Noise, sampling, and efficient genetic algorithms*. University of Illinois at Urbana-Champaign.
- Miller, B. L. & Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2):113--131.
- Miller, B. L.; Goldberg, D. E. et al. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193--212.
- Minkin, S. (1983). Assessing the quadratic approximation to the log likelihood function in nonnormal linear models. *Biometrika*, 70(2):367--372.
- Montero, E.; Riff, M.-C. & Neveu, B. (2014). A beginner's guide to tuning methods. *Applied Soft Computing*, 17:39--51.
- Montgomery, D. C. (2012). *Design and Analysis of Experiments*. Wiley, 8 edição.
- Mora, A. M.; Fernández-Ares, A.; Merelo, J. J.; García-Sánchez, P. & Fernandes, C. M. (2012). Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. *Journal of Computer Science and Technology*, 27(5):1007--1023.
- Muja, M. & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. Em *International Conference on Computer Vision Theory and Application VISSAPP'09*, pp. 331--340. INSTICC Press.
- Muja, M. & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.
- Müller, J.; Shoemaker, C. A. & Piché, R. (2013). SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Computers & Operations Research*, 40(5):1383--1400.
- Mulvey, J. M. et al. (1992). A diagonal quadratic approximation method for large scale linear programs. *Operations Research Letters*, 12(4):205--215.

- Nannen, V. & Eiben, A. (2007a). Efficient relevance estimation and value calibration of evolutionary algorithm parameters. Em *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 103--110. IEEE.
- Nannen, V. & Eiben, A. E. (2007b). Relevance estimation and value calibration of evolutionary algorithm parameters. Em *IJCAI*, volume 7, pp. 6--12.
- Nasayama, S.; Sasao, T. & Butler, J. T. (2006). Programmable numerical function generators based on quadratic approximation: architecture and synthesis method. Em *Design Automation, 2006. Asia and South Pacific Conference on*, pp. 6--pp. IEEE.
- Nelder, J. A. & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308--313.
- Nepomuceno, E. G. (2005). *Dinâmica, Modelagem e Controle de Epidemias*. Tese de doutorado, Programa de Pós-Graduação em Engenharia Elétrica.
- Neumaier, A. (1997). Molecular modeling of proteins and mathematical prediction of protein structure. *SIAM Review*, 39(3):407--460.
- Notsu, A.; Kane, S.; Ubukata, S. & Honda, K. (2016). Application of the UCT algorithm for noisy optimization problems. Em *Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems, 2016 Joint 8th International Conference on*, pp. 48--52. IEEE.
- Ong, Y. S.; Nair, P. B. & Keane, A. J. (2003). Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4):687--696.
- Ong, Y.-S.; Nair, P. B. & Lum, K. Y. (2006). Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, 10(4):392--404.
- Ostermeier, A.; Gawelczyk, A. & Hansen, N. (1994a). A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369--380.
- Ostermeier, A.; Gawelczyk, A. & Hansen, N. (1994b). Step-size adaptation based on non-local use of selection information. Em *International Conference on Parallel Problem Solving from Nature*, pp. 189--198. Springer.
- Pankov, A. R.; Platonov, E. & Semenikhin, K. V. (2003). Minimax optimization of investment portfolio by quantile criterion. *Automation and Remote control*, 64(7):1122--1137.

- Penalver, J. G. & Merelo, J. (1998). Optimizing web page layout using an annealed genetic algorithm as client-side script. Em *International Conference on Parallel Problem Solving from Nature*, pp. 1018--1027. Springer.
- Phelps, S. & Köksalan, M. (2003). An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12):1726--1738.
- Phillips, N. D. (2016). Yarr! the pirate's guide to R. <https://ndphillips.github.io/piratesguide.html>.
- Pierret, S. & Van den Braembussche, R. (1998). Turbomachinery blade design using a navier-stokes solver and artificial neural network. Em *ASME 1998 International Gas Turbine and Aeroengine Congress and Exhibition*, pp. V001T01A002--V001T01A002. American Society of Mechanical Engineers.
- Powell, M. J. (2002). UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555--582.
- Powell, M. J. (2003). On trust region methods for unconstrained minimization without derivatives. *Mathematical Programming*, 97(3):605--623.
- Powell, M. J. (2006). The NEWUOA software for unconstrained optimization without derivatives. Em *Large-scale nonlinear optimization*, pp. 255--297. Springer.
- Qian, C.; Yu, Y. & Zhou, Z.-H. (2015). Analyzing evolutionary optimization in noisy environments. *Evolutionary Computation*.
- Qu, X.; Zhou, J.; Song, E. & Zhu, Y. (2010). Minimax robust optimal estimation fusion in distributed multisensor systems with uncertainties. *IEEE Signal Processing Letters*, 17(9):811--814.
- Quade, D. (1979). Using weighted rankings in the analysis of complete blocks with additive block effects. *Journal of the American Statistical Association*, 74(367):680-683.
- Rakshit, P.; Konar, A. & Das, S. (2016). Noisy evolutionary optimization algorithms-a comprehensive survey. *Swarm and Evolutionary Computation*.
- Rakshit, P.; Konar, A.; Das, S.; Jain, L. C. & Nagar, A. K. (2014). Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(7):922--937.

- Ratle, A. (1998). Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. Em *International Conference on Parallel Problem Solving from Nature*, pp. 87--96. Springer.
- Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. Em *Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.* Ministry of Aviation, Royal Aircraft Establishment, Farnborough Hants.
- Rechenberg, I. (1973). *Evolutionsstrategie;: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann-Holzboog.
- Rekanos, I. T. (2008). Shape reconstruction of a perfectly conducting scatterer using differential evolution and particle swarm optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):1967--1974.
- Romero, J. J. & Machado, P. (2007). *The art of artificial evolution: a handbook on evolutionary art and music.* Springer Science & Business Media.
- Ros, R. & Hansen, N. (2008). A simple modification in CMA-ES achieving linear time and space complexity. Em *International Conference on Parallel Problem Solving from Nature*, pp. 296--305. Springer.
- Rosen, J. B. & Marcia, R. F. (2004). Convex quadratic approximation. *Computational Optimization and Applications*, 28(2):173--184.
- Rosenfelder, R. & Schreiber, A. W. (2001). On the best quadratic approximation in Feynman's path integral treatment of the polaron. *Physics Letters A*, 284(2):63--71.
- Rost, B. & Sander, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics*, 19(1):55--72.
- Samad, A.; Kim, K.-Y.; Goel, T.; Haftka, R. T. & Shyy, W. (2008). Multiple surrogate modeling for axial compressor blade shape optimization. *Journal of Propulsion and Power*, 24(2):301--310.
- Schneider, G.; Schuchhardt, J. & Wrede, P. (1994). Artificial neural networks and simulated molecular evolution are potential tools for sequence-oriented protein design. *Computer Applications in the Biosciences: CABIOS*, 10(6):635--645.
- Schwefel, H.-P. (1965). Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Dissertação de mestrado, Technical University of Berlin, Germany.

- Schwefel, H.-P. (1981). *Numerical optimization of computer models*. John Wiley & Sons, Inc.
- Shafer, R. (1974). On quadratic approximation. *SIAM Journal on Numerical Analysis*, 11(2):447--460.
- Shir, O. M.; Roslund, J.; Bäck, T. & Rabitz, H. (2008). Performance analysis of derandomized evolution strategies in quantum control experiments. Em *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 519-526. ACM.
- Siegmund, F.; Ng, A. H. & Deb, K. (2013). A comparative study of dynamic resampling strategies for guided evolutionary multi-objective optimization. Em *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 1826--1835. IEEE.
- Smith, J. R. & Schmidt, J. (1977). Optimization of simulation models using successive quadratic approximations. *Computers & Industrial Engineering*, 1(3):175--183.
- Storn, R. & Price, K. (1995). *Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces*, volume 3. ICSI Berkeley.
- Stroustrup, B. (2013). *The C++ programming language*. Pearson Education.
- Syberfeldt, A.; Ng, A.; John, R. I. & Moore, P. (2010). Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. *European Journal of Operational Research*, 204(3):533--544.
- Taghiyeh, S. & Xu, J. (2016). A new particle swarm optimization algorithm for noisy optimization problems. *Swarm Intelligence*, 10(3):161--192.
- Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275--1296.
- Takahashi, R. H. C. (2007). Otimização escalar e vetorial – volume 2. <http://www.mat.ufmg.br/~taka/>.
- Tanabe, R. & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. Em *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 1658--1665. IEEE.
- Van Thoai, N. (2005). General quadratic programming. Em *Essays and Surveys in Global Optimization*, pp. 107--129. Springer.

- Wan, Y.; Hu, G. & Dong, G. (2014). Randomized kd-trees based region growing for segmentation on curve surface. Em *Image and Signal Processing (CISP), 2014 7th International Congress on*, pp. 447--451. IEEE.
- Wang, H.; Zhu, X. & Du, Z. (2010). Aerodynamic optimization for low pressure turbine exhaust hood using kriging surrogate model. *International Communications in Heat and Mass Transfer*, 37(8):998--1003.
- Wang, Y. (2007). Maximum likelihood computation based on the Fisher scoring and Gauss-Newton quadratic approximations. *Computational Statistics & Data Analysis*, 51(8):3776--3787.
- Wanner, E. F.; Guimarães, F. G.; Saldanha, R. R.; Takahashi, R. H. & Fleming, P. J. (2005). Constraint quadratic approximation operator for treating equality constraints with genetic algorithms. Em *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pp. 2255--2262. IEEE.
- Wanner, E. F.; Guimaraes, F. G.; Takahashi, R. H. & Fleming, P. J. (2006a). Quadratic approximation-based coordinate change in genetic algorithms. Em *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 1960--1967. IEEE.
- Wanner, E. F.; Guimaraes, F. G.; Takahashi, R. H. & Fleming, P. J. (2006b). A quadratic approximation-based local search procedure for multiobjective genetic algorithms. Em *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 938--945. IEEE.
- Wanner, E. F.; Guimaraes, F. G.; Takahashi, R. H. & Fleming, P. J. (2007). Local search with quadratic approximation in genetic algorithms for expensive optimization problems. Em *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 677--683. IEEE.
- Wanner, E. F.; Guimarães, F. G.; Takahashi, R. H. & Fleming, P. J. (2008a). Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. *Evolutionary Computation*, 16(2):185--224.
- Wanner, E. F.; Guimares, F. G.; Takahashi, R. H.; Lowther, D. A. & Ramirez, J. A. (2008b). Multiobjective memetic algorithms with quadratic approximation-based local search for expensive optimization in electromagnetics. *IEEE Transactions on Magnetism*, 44(6):1126--1129.

- Werner, W. (1986). Iterative solution of systems of nonlinear equations based upon quadratic approximations. *Computers & Mathematics with Applications*, 12(3):331-343.
- Xie, J.; Nie, Y. M. & Yang, X. (2013). Quadratic approximation and convergence of some bush-based algorithms for the traffic assignment problem. *Transportation Research Part B: Methodological*, 56:15--30.
- Xu, S.; Li, X. & Lu, W. F. (2016). Randomized kd tree ReliefF algorithm for feature selection in handling high dimensional process parameter data. Em *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pp. 1--8. IEEE.
- Xue, A.; Wu, F. F.; Ni, Y.; Mei, S. & Lu, Q. (2006). Power system transient stability assessment based on quadratic approximation of stability region. *Electric Power Systems Research*, 76(9):709--715.
- Yang, D. & Flockton, S. J. (1995). Evolutionary algorithms with a coarse-to-fine function smoothing. Em *Evolutionary Computation, 1995., IEEE International Conference on*, volume 2, pp. 657--662. IEEE.
- Yao, X. & Liu, Y. (1997). A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3):694--713.
- Zarifia, M. H.; Ghalehjogh, N. K. & Baradaran-nia, M. (2015). A new evolutionary approach for neural spike detection based on genetic algorithm. *Expert Systems with Applications*, 42(1):462--467.
- Zhao, X.-L.; Wang, F. & Ng, M. K. (2014). A new convex optimization model for multiplicative noise and blur removal. *SIAM Journal on Imaging Sciences*, 7(1):456-475.
- Zhou, Y.; Endres, C. J.; Brašić, J. R.; Huang, S.-C. & Wong, D. F. (2003). Linear regression with spatial constraint to generate parametric images of ligand-receptor dynamic PET studies with a simplified reference tissue model. *Neuroimage*, 18(4):975-989.

Apêndice A

Sintonização dos Parâmetros de Algoritmos

A.1 Apresentação

O processo de ajuste dos parâmetros de um algoritmo de otimização, que busca otimizar um critério de qualidade (função de utilidade) desse algoritmo é aqui denominado *sintonização de parâmetros*. Para maiores informações sobre a temática da sintonização de parâmetros, pode-se consultar as referências [Hutter et al., 2009; Birattari et al., 2010; Eiben & Smit, 2011; Montero et al., 2014].

Neste apêndice, é apresentada uma heurística de sintonização de parâmetros que foi desenvolvida no âmbito deste trabalho de tese, denominada *Robust Parameter Searcher* (RPS), que visa determinar um conjunto fixo de parâmetros que corresponde a um ajuste adequado do algoritmo para que este otimize de maneira eficiente uma base de problemas que se considera representativa de uma classe de problemas de interesse. Um exemplo de aplicação da heurística RPS é apresentado, considerando o caso do ajuste de parâmetros de algoritmos da classe da Evolução Diferencial (DE). Nesse exemplo, é também realizada a comparação dos resultados obtidos pela heurística RPS com os resultados oriundos da aplicação de vários outros procedimentos de sintonização de parâmetros descritos na literatura. Os resultados dessa comparação sugerem a superioridade da heurística aqui proposta.

No contexto desta tese, a heurística de sintonização de parâmetros RPS é empregada para obter os conjuntos de parâmetros a serem empregados nos algoritmos ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC considerando a base dos problemas apresentados na Tabela 4.1 em ambientes sem e com ruído na avaliação. Ajustes de parâmetros

distintos foram considerados para o caso com ruído e para o caso sem ruído, de forma a permitir o estudo dos algoritmos nas condições de sua melhor performance. Sem uma heurística do tipo RPS, o estudo desenvolvido no Capítulo 4 desta tese não teria sido possível.

A.2 Definição do Problema de Sintonização de Parâmetros

Seja \mathcal{A} um algoritmo e $\Lambda(\boldsymbol{\pi}, f, \zeta, \kappa)$ uma instância de \mathcal{A} para a qual $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_m) \in \Pi$ representa o vetor de m parâmetros de Λ que constitui o conjunto de variáveis de decisão do problema de sintonização de parâmetros, sendo Π a região factível para busca desses parâmetros. $f : \mathbb{R}^n \rightarrow \mathbb{R}, f \in \mathcal{F}$, é uma função-objetivo que deve ser minimizada por Λ , que visa encontrar a solução ótima $\mathbf{x} = \mathbf{x}^*$ na região factível $\mathcal{S} \subseteq \mathbb{R}^n$ do espaço de variáveis de decisão de $f(\mathbf{x})$, e \mathcal{F} representa a classe de funções de interesse, para a qual deseja-se sintonizar os parâmetros de \mathcal{A} . $\zeta \in \mathbb{N}$ é um número natural que representa a semente para o gerador de números aleatórios de Λ (o que significa que duas execuções de Λ com o mesmo conjunto de entradas $(\boldsymbol{\pi}, f, \zeta, \kappa)$ conduzirão ao mesmo resultado). $\kappa \in \mathbb{N}^*$ é um orçamento computacional, medido aqui em número de avaliações de função-objetivo disponível para uma execução de Λ sobre $f(\mathbf{x})$. Considere-se o espaço $\Omega \subseteq \Pi \times \mathcal{F} \times \mathbb{N} \times \mathbb{N}^*$ de todas as possíveis entradas para Λ . Para cada $\omega \in \Omega$, uma execução de $\Lambda(\omega)$ determina um único melhor valor de função-objetivo encontrado $\gamma = f(\mathbf{x}^b)$ associado à melhor solução encontrada $\mathbf{x}^b \in \mathcal{S}$.

No *problema de sintonização de parâmetros* aqui definido, deseja-se obter os melhores parâmetros $\boldsymbol{\pi} \in \Pi$ do algoritmo \mathcal{A} para um dado conjunto de t funções-objetivo de treinamento $f : \mathbb{R}^n \rightarrow \mathbb{R} \in \mathcal{F} = \{f_1, f_2, \dots, f_t\}$, considerando um orçamento computacional fixo κ para cada execução de Λ e uma determinada *função de utilidade* ν . Neste apêndice, as menções a *funções-objetivo*, ou simplesmente *objetivos*, se referirão sempre a funções-objetivo dos algoritmos cujos parâmetros estão sendo sintonizados, enquanto as menções a *funções de utilidade* se referirão às funções consideradas pela heurística cuja tarefa é a de sintonizar os parâmetros dos algoritmos de otimização. Deve-se notar que, como simplificação do problema mais geral de sintonização dos parâmetros de \mathcal{A} , não se discute aqui a questão de como escolher um conjunto $\{f_1, f_2, \dots, f_t\}$ que seja representativo da classe de funções de interesse \mathcal{F} , sendo adotada a simplificação de supor que $\mathcal{F} = \{f_1, f_2, \dots, f_t\}$.

Seja um objetivo $f_i, i \in \{1, \dots, t\}$, e seja $\zeta_j \in \{\zeta_1, \dots, \zeta_{e_i}\}$ para um inteiro e_i .

Define-se:

$$\gamma_{i,j}(\boldsymbol{\pi}) = \Lambda(\boldsymbol{\pi}, f_i, \zeta_j, \kappa) \quad (\text{A.1})$$

A função $\gamma_{i,j}(\boldsymbol{\pi})$ representa o resultado da execução do algoritmo \mathcal{A} , instanciado em $\Lambda(\boldsymbol{\pi}, f_i, \zeta_j, \kappa)$, considerando o conjunto de parâmetros $\boldsymbol{\pi}$, o objetivo f_i e a semente do gerador de números aleatórios ζ_j , sendo o valor de κ considerado um parâmetro constante cujo valor é preestabelecido. Define-se a estatística:

$$\hat{\gamma}_i = \frac{1}{e_i} \sum_{j=1}^{e_i} \gamma_{i,j} \quad (\text{A.2})$$

que corresponde a um estimador da esperança de $\gamma_{i,j}(\boldsymbol{\pi})$ sobre todos os possíveis valores de ζ_j .

Considerando todas as t funções-objetivo, define-se a *Mean Best Fitness* (MBF):

$$\nu(\hat{\gamma}_1, \hat{\gamma}_2, \dots, \hat{\gamma}_t) = \frac{\sum_{i=1}^t \hat{\gamma}_i}{t} \quad (\text{A.3})$$

A Equação (A.4) apresenta a formalização do problema de sintonização de parâmetros que resulta no melhor parâmetro $\boldsymbol{\pi}$ para uma instância Λ do algoritmo \mathcal{A} ao otimizar o conjunto de funções \mathcal{F} com um orçamento de κ avaliações em cada execução de Λ com distintas sementes aleatórias ζ .

$$\boldsymbol{\pi}^* = \arg \min_{\boldsymbol{\pi}} \nu(\hat{\gamma}_1, \hat{\gamma}_2, \dots, \hat{\gamma}_t) \quad (\text{A.4})$$

Sobre o problema apresentado pela Equação (A.4), pode-se notar que trata-se de problema computacionalmente caro, uma vez que cada avaliação da função de utilidade ν envolve, por sua vez, um número κ de avaliações de cada uma das funções-objetivo f_i . Essa função de utilidade trata-se de função intrinsecamente ruidosa, em virtude da execução de avaliações de função-objetivo ser realizada sempre com distintas sementes para o gerador de números aleatórios. Além disso, pode-se esperar que ν possa ser multimodal. Desta forma, conclui-se que uma heurística de sintonização de parâmetros deva ser, pelo menos, capaz de lidar com funções de utilidade ruidosas e que consiga retornar boas soluções para o problema de sintonização de parâmetros dispondo, globalmente, de um orçamento relativamente limitado de avaliações de função-objetivo.

A.3 *Robust Parameter Searcher*

Foi aqui desenvolvida a heurística denominada *Robust Parameter Searcher*, ou simplesmente RPS, que é baseada no algoritmo Nelder Mead Simplex [Nelder & Mead, 1965; Chang, 2012] com restrições de limite de variáveis, um operador de comparação baseado em confiança, e reavaliação de soluções do simplex, com número máximo que cresce linearmente (até um limite) com a quantidade de avaliações de função globalmente realizadas.

O RPS, descrito na Figura A.1, possui sete parâmetros configuráveis: $\delta_e \in [1,1; 3,0]$ é o coeficiente de expansão; $\delta_c \in [0,1; 0,9]$ é o coeficiente de contração; $\delta_s \in [0,1; 0,9]$ é o coeficiente de encolhimento do simplex; $\mathbf{f}_p \in [0,001; 0,100]$ é a fração para determinar o raio inicial, para cada variável que representa um parâmetro, de acordo com os respectivos limites inferior e superior; $\varepsilon \in [1,0; 0,001]$ é a distância Euclidiana limiar permitida entre a melhor e a pior solução, usada no critério de reinicialização do simplex; $e_{\max} \in \{2; \dots; 10\}$ é o número máximo de reavaliações de uma solução no final da execução da heurística; $p_\alpha \in [0,05; 0,5]$ é o nível de significância para comparar pares de soluções em uma comparação baseada em confiança estatística. Há também, implicitamente, um parâmetro que o usuário deve configurar previamente, que é o número de casas decimais que a representação das variáveis pode conter. Desta forma, cada operador que gera ou altera soluções faz um arredondamento considerando esta quantidade de casas decimais. No caso de parâmetros discretos, estes são arredondados para baixo.

Qualquer solução é caracterizada por um vetor de parâmetros $\boldsymbol{\pi}$, um vetor de avaliações de função de utilidade \mathbf{u} , e uma variável binária b que indica factibilidade em relação às restrições de limite, analisada no momento da criação do vetor de variáveis de decisão. Note-se que uma avaliação da função utilidade ν implica em $t \times \kappa$ avaliações das funções-objetivo f_i .

Em qualquer iteração, uma solução (um vetor $\boldsymbol{\pi}$) pode ter sua função utilidade avaliada no máximo e vezes. Tal número cresce linearmente de acordo com a quantidade corrente de avaliações \mathfrak{e} , conforme Equação (A.5), em que \mathfrak{E} é o máximo de avaliações de função utilidade permitidas. Inicialmente, são permitidas duas repetições e, ao final, o limite é e_{\max} .

$$e = 1 + (e_{\max} - 1) \left\lfloor \frac{\mathfrak{e} + 0,5}{\mathfrak{E}} \right\rfloor \quad (\text{A.5})$$

Na comparação entre um par de soluções, consideram-se as seguintes regras:

- (i) Vetores infactíveis não são avaliados, de modo que quando ambos são infactíveis,

```

Entrada:  $\delta_e, \delta_c, \delta_s, \mathbf{f}_p, \varepsilon, \mathbf{e}_{\max}, p_\alpha$ 
Dados:  $\mathcal{A}, \Pi, \mathcal{F}, \zeta, \kappa, \mathfrak{E}$ 
1 início
2    $m \leftarrow NumMetaParametros(\Pi);$ 
3    $[i_w, i_c, i_r, i_e, i_{oc}, i_{ic}] \leftarrow [m + 1, m + 2, m + 3, m + 4, m + 4, m + 5];$ 
4    $\epsilon \leftarrow 1;$ 
5    $[S, \epsilon, \mathbf{e}] \leftarrow CriaAvalSimplex(\Pi, \mathbf{f}_p, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
6   enquanto  $\epsilon \leq \mathfrak{E}$  faça
7     se  $\|S[1].\boldsymbol{\pi} - S[m + 1].\boldsymbol{\pi}\| \leq \varepsilon$  então
8        $[S, \epsilon, \mathbf{e}] \leftarrow ReiniciaAvalSimplex(S, \Pi, \mathbf{f}_p, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
9       fim
10       $[S, \epsilon, \mathbf{e}] \leftarrow OrdenaSimplex(S, m, \mathbf{e}, \mathbf{e}_{\max}, p_\alpha);$ 
11      se  $\epsilon \geq \mathfrak{E}$  então saia do enquanto;
12       $S[i_c] \leftarrow CriaCentroide(S);$ 
13       $[S[i_r], \epsilon, \mathbf{e}] \leftarrow CriaAvalReflecao(S, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
14      se  $S[1] \stackrel{p_\alpha}{\leq} S[i_r] < S[m]$  então  $S[i_w] \leftarrow S[i_r];$  continue;
15      se  $S[i_r] < S[1]$  então
16         $[S[i_e], \epsilon, \mathbf{e}] \leftarrow CriaAvalExpansao(S, \delta_e, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
17        se  $S[i_e] \stackrel{p_\alpha}{<} S[i_r]$  então  $S[i_w] \leftarrow S[i_e];$  senão  $S[i_w] \leftarrow S[i_r];$ 
18        continue;
19      fim
20       $shrink \leftarrow \text{falso};$ 
21      se  $S[m] \stackrel{p_\alpha}{\leq} S[i_r] < S[i_w]$  então
22         $[S[i_{oc}], \epsilon, \mathbf{e}] \leftarrow CriaAvalContExterna(S, \delta_c, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
23        se  $S[i_{oc}] \stackrel{p_\alpha}{\leq} S[i_r]$  então  $S[i_w] \leftarrow S[i_{oc}];$  continue;
24      senão
25         $shrink \leftarrow \text{verdadeiro};$ 
26      fim
27      se  $\neg shrink$  então
28        se  $S[i_r] \stackrel{p_\alpha}{\geq} S[i_w]$  então
29           $[S[i_{ic}], \epsilon, \mathbf{e}] \leftarrow CriaAvalContInterna(S, \delta_c, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
30          se  $S[i_{ic}] \stackrel{p_\alpha}{<} S[i_w]$  então  $S[i_w] \leftarrow S[i_{ic}];$  continue;
31        fim
32      fim
33       $[S, \epsilon, \mathbf{e}] \leftarrow ContraiAvalSimplex(S, m, \delta_e, \epsilon, \mathfrak{E}, \mathcal{F}, \zeta, \kappa, \mathbf{e}, \mathbf{e}_{\max});$ 
34    fim
35     $\boldsymbol{\pi} \leftarrow S[1].\boldsymbol{\pi};$ 
36     $F_u \leftarrow MBF(S[1].\mathbf{u});$ 
37 fim
38 retorna  $[\boldsymbol{\pi}, F_u]$ 

```

Figura A.1: Algoritmo *Robust Parameter Searcher* (RPS) para sintonização de parâmetros de algoritmos de otimização.

compara-se o somatório dos níveis de violação. É considerado melhor aquele que infringiu menos as restrições. (ii) Quando uma solução é factível e a outra não, considera-se melhor a primeira. (iii) Quando ambas são factíveis, o vetor de avaliações é testado de acordo com um teste-t em um nível de significância p_α . Se foi observada diferença significativa, então a comparação é concluída. Senão, avalia-se aquela com maior desvio padrão amostral que não tenha atingido o limite e . Se tal limitação desta solução já foi atingida e a da outra não, então avalia-se a segunda. Tal processo se repete até seja observada diferença significativa ou o limite de avaliações para ambas seja atingido. Caso o número total de reavaliações para ambas tenha atingido o máximo, a comparação é realizada via média amostral. Este operador é representado por um símbolo de comparação relacional indicativo com um p_α sobrescrito. A atualização de e e e foi deixada implícita por questão de espaço. Embora não sejam garantidas as premissas para o uso do teste-t [Montgomery, 2012], a heurística de comparação proposta pode melhorar a qualidade da tomada decisão na presença de ruído.

Inicialmente, no RPS, extrai-se o número de parâmetros, m , da definição de região de busca Π . Em seguida, são definidos para melhorar a legibilidade os índices que armazenarão a pior solução do simplex (i_w), a solução centroide (i_c), a solução refletida (i_r), a expandida (i_e), a contraída externamente (i_{oc}) e a contraída internamente (i_{ic}).

A função *CriaAvalSimplex* gera e avalia o primeiro simplex com $m+1$ vetores de parâmetros. Utilizando um ponto inicial escolhido aleatoriamente com probabilidade uniforme dentro de Π e uma base vetorial qualquer, aplica-se a transformação de Householder [Householder, 1958] para se criar uma base ortonormal. Assim, pondera-se cada variável deste conjunto com $m+1$ soluções contidas no simplex pelo raio inicial $r_0[i]$, conforme Equação (A.6). Nesta, $L_\Pi[i]$ e $U_\Pi[i]$ são os limites inferior e superior para a i -ésima variável do vetor π . Em seguida, avaliam-se estas primeiras $m+1$ soluções em S , considerando \mathcal{F} , ζ e κ . O vetor S e o número corrente de avaliações e são retornados.

$$r_0[i] = \mathbf{f}_p \cdot (U_\Pi[i] - L_\Pi[i]), \quad i = 1, 2, \dots, m. \quad (\text{A.6})$$

Em seguida, inicia-se o laço de repetição que termina quando o número máximo de avaliações é atingido. Primeiro, é verificado se a distância entre a melhor e a pior solução do simplex está abaixo do limiar ε . Caso verdadeiro, considera-se que houve estagnação e uma reinicialização é executada de modo que se mantém a melhor solução e outras m novas soluções são geradas via transformação de Householder, sendo cada variável ponderada por $r_\circ[i]$, $\forall i \in \{1, 2, \dots, m\}$.

O valor de $r_\circ[i]$ é igual a $r_0[i]$ no caso de a melhor solução corrente ser distinta da

melhor no instante da última reinicialização do simplex. Sempre que a melhor solução corrente não se altera, $r_{\circ}[i]$ acumula o valor de $r_0[i]$. Tal mecanismo, além de criar novas soluções ortogonais, permite que o raio cresça de modo a escapar de ótimos locais. Após sua geração, tais m novos vetores são avaliados.

O próximo passo é ordenar crescentemente as $m + 1$ soluções de acordo com a função de utilidade, através do operador $\stackrel{p_{\alpha}}{<}$, que possibilita reavaliações. A seguir, é verificado se a quantidade de avaliações atingiu o limite. Em caso positivo, o fluxo de execução sai do laço de repetição e o melhor vetor encontrado é retornado. Caso contrário, a iteração prossegue.

Na função *CriaCentroide*, cria-se a solução centróide usando as primeiras m soluções, de acordo com a Equação (A.7), que não será avaliada, mas usada na criação de outros vetores. Logo após, usando a Equação (A.8), a solução refletida no oposto da pior é criada e posteriormente avaliada em *CriaAvalReflecao*. Se esta for melhor que a m -ésima e não preferível em relação à melhor corrente, então ela assume o lugar da pior do simplex, e o fluxo retorna para o início do laço de repetição. Em seguida, se a solução refletida é mais vantajosa do que a melhor solução corrente, então uma solução expandida é criada, via Equação (A.9), e avaliada em *CriaAvalExpansao*. No caso de esta nova solução ser melhor do que a solução refletida, ela será aceita no lugar da pior solução corrente; em caso contrário a solução refletida entrará no simplex. A seguir, o laço de repetição é retomado do início.

$$S[i_c].\boldsymbol{\pi} = \frac{1}{m} \sum_{j=1}^m S[j].\boldsymbol{\pi} \quad (\text{A.7})$$

$$S[i_r].\boldsymbol{\pi} = S[i_c].\boldsymbol{\pi} + (S[i_c].\boldsymbol{\pi} - S[i_w].\boldsymbol{\pi}) \quad (\text{A.8})$$

$$S[i_e].\boldsymbol{\pi} = S[i_c].\boldsymbol{\pi} + \delta_e \cdot (S[i_r].\boldsymbol{\pi} - S[i_c].\boldsymbol{\pi}) \quad (\text{A.9})$$

A seguir, verifica-se se a reflexão é melhor que a $(m + 1)$ -ésima solução, não sendo entretanto melhor que a m -ésima solução. Em caso negativo, a próxima ação do algoritmo será encolher o simplex. Em caso positivo, uma contração externa é criada, via Equação (A.10), e avaliada em *CriaAvalContExterna*. Se esta contração for melhor que a refletida, ela será aceita no lugar da pior solução no simplex.

$$S[i_{oc}].\boldsymbol{\pi} = S[i_c].\boldsymbol{\pi} + \delta_c \cdot (S[i_r].\boldsymbol{\pi} - S[i_c].\boldsymbol{\pi}) \quad (\text{A.10})$$

Na sequência, se não foi sinalizado que o simplex deva ser encolhido e a reflexão não é melhor que a pior solução, então a função *CriaAvalContInterna* cria um vetor

contraído internamente, usando a Equação (A.11), e o avalia. Caso esta contração seja melhor que a pior solução corrente, então ocorre a substituição e prossegue-se para uma nova iteração.

$$S[i_c].\pi = S[i_r].\pi - \delta_c \cdot (S[i_r].\pi - S[i_c].\pi) \quad (\text{A.11})$$

Se nenhuma das tentativas anteriores foi aceita, então a função *ContraiaAvalSimplex* é executada. Nesta, os m novos vetores são encolhidos em torno da melhor solução corrente, utilizando a Equação (A.12), e avaliados em seguida. Ao final do algoritmo, fora do laço de repetição, o melhor vetor, com a respectiva MBF encontrados, são retornados.

$$S[j].\pi = S[1].\pi + \delta_s \cdot (S[j].\pi - S[1].\pi), \quad j = 2, \dots, m + 1. \quad (\text{A.12})$$

A.4 Comparações com Outras Heurísticas de Sintonização

Apresenta-se agora nesta seção o planejamento, execução e análise de uma bateria de experimentos computacionais que comparam o RPS com outras cinco heurísticas de sintonização de parâmetros de algoritmos, que são: Random Search (RS), Parameter Iterated Local Search (ParamILS) [Hutter et al., 2007, 2009], Algoritmo Genético (mGA), Relevance Estimation and Value Calibration (Revac) [Nannen & Eiben, 2007b,a] e Bounded Nelder Mead (bNM). Da mesma forma que o RPS, a implementação desses métodos arredonda as variáveis para uma precisão previamente estabelecida. No estudo de caso apresentado, será considerada a busca de parâmetros reais com precisão de 6 casas decimais. Entretanto, diferentemente do algoritmo proposto, tais heurísticas comparativas avaliam apenas uma vez cada vetor e fazem comparação pareada usando o operador relacional “menor que” sobre tais únicos valores de utilidade.

A.4.1 Algoritmos Comparados

A estratégia RS, usada como base de comparação, não possui parâmetro de entrada e executa uma busca aleatória para determinar o melhor conjunto de parâmetros possível para uma heurística. Inicialmente, gera-se uma solução dentro do espaço de busca, que é avaliada. Ulteriormente, enquanto o número de avaliações total não é alcançado, sorteia-se aleatoriamente de maneira uniforme um novo vetor sobre a região de busca,

que é avaliado e comparado com a melhor solução até então encontrada. Caso seja melhor, esta assumirá o posto de melhor vetor.

O ParamILS [Hutter et al., 2007] é uma busca local iterativa que possui três parâmetros, que são $ils_1 \in [0,01; 0,5]$, o percentual em relação ao total de avaliações dedicado a uma busca aleatória inicial; $ils_2 \in \{1; \dots 100\}$, o número total de perturbações na solução corrente no início do laço evolutivo, considerando todas as variáveis; e $ils_3 \in [0,001; 0,1]$, a probabilidade de reinicializar a solução corrente na região de busca. Primeiramente, gasta-se o percentual ils_1 do total de avaliações para se determinar o vetor corrente inicial usando o algoritmo RS. Depois, sobre este é executado o *iterative first improvement*, que visita e avalia todos os vizinhos (diferem em uma variável pela precisão originalmente dada), retornando a melhor solução como solução corrente. No laço de repetição, que finda ao avaliar um total de conjunto de parâmetros, ocorrem as seguintes etapas: (i) Geração aleatória e consecutiva de ils_2 vizinhos a partir da solução corrente, sendo o tamanho do passo dado pela precisão dada; (ii) Sobre o vetor final gerado realiza-se o *iterative first improvement*, sendo que caso a solução desta operação seja melhor que a corrente, a primeira assume o posto de solução corrente; (iii) A seguir, com probabilidade ils_3 , gera-se aleatoriamente uma nova solução corrente. Durante todo este processo evolutivo pratica-se o elitismo, sendo armazenada a melhor solução encontrada.

O mGA é uma heurística de sintonização baseada em algoritmo genético com os parâmetros $mga_1 \in \{4; \dots; 300\}$, o tamanho da população de soluções; $mga_2 \in [0,05; 0,95]$, a taxa de elitismo usada no processo seletivo para a passagem de geração; $mga_3 \in [0,01; 0,50]$, a probabilidade de mutação; e $mga_4 \in [0,01; 0,90]$, a probabilidade de cada variável ser mutada quando a ocorre a mutação. No começo, gera-se e avalia-se a primeira população de mga_1 pais. Dentro do laço de repetição evolutivo ocorre: (i) A seleção via roleta, cuja aptidão linear é dada pela ordenação dos valores da função de utilidade; (ii) A geração de novos mga_1 filhos dentro da hipercaixa determinada por cada par de pais selecionado, através do cruzamento, e mutação a uma taxa mga_3 alterando cada variável a uma probabilidade de mga_4 , sorteando um valor dentro da região de busca; (iii) A avaliação dos filhos; e (iv) A seleção dos novos vetores pais para a próxima geração é feita a partir do conjunto união das populações de pais e filhos, sendo selecionadas as $\lfloor mga_1 \cdot mga_2 \rfloor$ melhores soluções de tal conjunto, e depois então escolhidas as $mga_1 - \lfloor mga_1 \cdot mga_2 \rfloor$ restantes de forma aleatória. Este processo se repete enquanto o total de avaliações da função de utilidade não é atingido.

O Revac [Nannen & Eiben, 2007b,a] é uma heurística baseada em algoritmo de estimação de distribuição com quatro parâmetros, sendo eles $revac_1 \in \{4; \dots; 300\}$, a quantidade de vetores na população; $revac_2 \in [0,1; 0,9]$ é o percentual, em relação

a $revac_1$, de soluções antigas a serem substituídas na população; $revac_3 \in [0,05; 0,50]$ é o percentual de melhores soluções a serem selecionadas e usadas no cruzamento; e $revac_4 \in [0,01; 0,50]$ é o fator de distância a ser usado para selecionar vetores durante o processo de mutação. Preliminarmente, define-se o número de trocas de soluções por iteração ($t_{rvc} = \lfloor revac_1 \cdot revac_2 \rfloor$), a quantidade de vetores usada no cruzamento ($q_{rvc} = \lfloor revac_1 \cdot revac_3 \rfloor$) e o limiar de distância a ser usado no processo de mutação ($l_{rvc} = \lfloor revac_1 \cdot revac_4 \rfloor$). Assim como no mGA, a população inicial é criada aleatoriamente e avaliada. Posteriormente, no laço que se repete até atingir o máximo de avaliações da função de utilidade, os passos a seguir são executados: (i) Seleção dos q_{rvc} melhores vetores e uso dos mesmos para gerar t_{rvc} novos vetores, em um cruzamento multiparental com pesos aleatórios uniformes e convexos; (ii) Mutação dos novos vetores; (iii) Avaliação dos novos parâmetros; e (iv) Substituição das novas soluções pelas mais antigas. O delineamento da mutação segue os seguintes passos: (a) Cópia das soluções para uma matriz auxiliar (cada linha representa uma variável); (b) Ordenação crescente de cada linha desta matriz; (c) Para cada variável dos novos vetores é atribuído uma valor aleatório uniforme cujos limites inferior e superior são dados pelos l_{rvc} -ésimos valores à esquerda e direita na linha da coluna da matriz a partir do valor obtido no cruzamento. Ao final, retorna-se o melhor conjunto de parâmetros encontrado durante todo o processo.

O bNM é uma implementação da heurística Nelder Mead Simplex [Nelder & Mead, 1965; Chang, 2012] com tratamento de restrições para limitar os valores das variáveis dentro da região de busca. Os parâmetros considerados são $bnm_1 \in [1,1; 3,0]$, o coeficiente de expansão; $bnm_2 \in [0,1; 0,9]$, o coeficiente de contração; $bnm_3 \in [0,1; 0,9]$ o fator de encolhimento do simplex; $bnm_4 \in [0,001; 0,1]$, a fração para determinar o raio inicial, para cada variável, em relação aos limites da região de busca; e $bnm_5 \in [0,001; 0,1]$, a distância Euclidiana mínima entre a melhor e a pior solução no simplex para reinicializá-lo. O bNM se diferencia do RPS pelo fato de não reavaliar um vetor, ou seja, permite-se apenas uma avaliação de utilidade para cada solução, que é usada na comparação da qualidade das soluções. Parâmetros inactíveis não são avaliados e recebem um valor que representa o nível de inactibilidade, dado pelo somatório das diferenças violadas ao quadrado por variável. Uma solução é considerada melhor que a outra se (i) ambas são inactíveis, mas a primeira possui um menor nível de inactibilidade; (ii) a primeira é factível e a outra não; ou (iii) ambas são factíveis e a primeira possui melhor valor de função de utilidade. Inicialmente, no bNM é criado um simplex inicial com vetores aleatórios, perpendiculares (via transformação de Householder [Householder, 1958]) e ponderados usando bnm_3 , que é posteriormente avaliado. No laço de repetição, que finda quando as avaliações atingirem um limite

dado, é testado se a diferença entre a melhor e a pior solução do simplex distam menos que bnm_4 . Em caso positivo, o simplex é reinicializado da mesma forma que no RPS. Em seguida, as operações usuais do Nelder Mead são realizadas, como a ordenação do simplex, criação e avaliação (quando adequado) dos vetores centróide, refletido, expandido, contraído externamente e internamente, e redução do simplex.

A.4.2 Estudo de Caso

Para comparar a eficiência do RPS em relação às estratégias RS, ParamILS, mGA, Revac e bNM, uma bateria de 5 experimentos computacionais foram planejados para ajustar os parâmetros de instâncias da Evolução Diferencial (DE) [Storn & Price, 1995; Das & Suganthan, 2011] que devem otimizar as 12 funções-objeto sem ruído do conjunto \mathcal{F} apresentado pela Tabela 4.1, na dimensão 10, com 1000 avaliações por execução. Cada experimento se diferencia dos demais pelo orçamento de avaliações de função utilidade, sendo consideradas 500, 2000, 3000, 5000 e 10000 avaliações.

Antes da bateria de testes, para definir um conjunto de parâmetros para o ParamILS, mGA, Revac, bNM e RPS sob um mesmo ambiente ruidoso, um pré-experimento também foi concebido. Este procedimento, embora recomendado pela literatura especializada [Eiben & Smit, 2011], nunca foi realizado para se comparar heurísticas de sintonização de parâmetros. Nas principais referências conhecidas, foram utilizados parâmetros configurados manualmente, como nos trabalhos de Nannen & Eiben [2007a] e de Hutter et al. [2009].

Para o treinamento das heurísticas de sintonização de parâmetros, estas são sintonizadas considerando a função de utilidade MBF, ν , construída de maneira artificial, considerando o subconjunto $\mathcal{F}_{mt} = \{\text{Elliptic, Rosenbrock, Rastrigin}\} \subset \mathcal{F}$ de funções avaliadas e somadas a um ruído Gaussiano com média 0 e desvio padrão 3 como se estas fossem representações dos algoritmos a serem sintonizados. Para o cômputo da ν , apenas uma execução para cada objetivo em \mathcal{F}_{mt} é realizada. A tarefa do ajuste dos parâmetros das heurísticas de sintonização, assim enquadradas como heurísticas de otimização de tais funções, é realizada por uma instância do mGA com parâmetros $(mga_1; mga_2; mga_3; mga_4) = (50; 0,2; 0,1; 0,5)$ que soluciona a Equação (A.4), com $\kappa = 5000$ avaliações, e encontra o respectivo $\pi_{\mathcal{A}}$ para cada $\mathcal{A} \in \{\text{ParamILS, mGA, Revac, bNM, RPS}\}$, com um orçamento de $\kappa_{\mathcal{A}} = 500$ avaliações por objetivo em cada execução.

Para cada algoritmo \mathcal{A} , o mGA é executado 100 vezes na configuração exposta pelo parágrafo anterior. A precisão adotada foi de 6 casas decimais. Parâmetros discretos são arredondados para baixo. Assim, obtém-se 100 candidatos a parâmetros

Tabela A.1: Conjunto de parâmetros ajustados para as heurísticas ParamILS, mGA, Revac, bNM e RPS.

π_{ParamILS}	=	(0,472171; 70; 0,009071)
π_{mGA}	=	(39; 0,890494; 0,392209; 0,399264)
π_{Revac}	=	(85; 0,610822; 0,077699; 0,445252)
π_{bNM}	=	(2,498700; 0,601812; 0,527752; 0,084575; 0,081973)
π_{RPS}	=	(2,667890; 0,687594; 0,522558; 0,097288; 0,091461; 9; 0,870956)

que serão ordenados, sendo deles extraídos os 20 melhores, de acordo com a melhor MBF retornada. Cada uma dessas 20 instâncias, representada pelo respectivo conjunto de parâmetros, será simulada novamente 100 vezes. Daí, cada conjunto candidato de parâmetros será representado por 100 amostras de melhores MBF retornadas, em relação a \mathcal{F}_{mt} . Na seleção do conjunto de parâmetros final, realiza-se o teste de Quade, com 95% de confiança e procedimento de Holm para ajustar o nível de significância na comparação pareada [Conover, 1999; García et al., 2010], para testar a hipótese sobre a igualdade das medianas de MBF para cada conjunto de parâmetros candidatos. Havendo empate na primeira colocação (não ocorrência de diferença significativa entre os primeiros colocados), aquele conjunto com menor mediana amostral é escolhido.

Após a execução dessa etapa preliminar, são obtidos os conjuntos de parâmetros π_{ParamILS} , π_{mGA} , π_{Revac} , π_{bNM} e π_{RPS} apresentados na Tabela A.1. Os gráficos beanplot e tabelas sobre estatísticas e teste de Quade foram omitidos por questão de espaço. Tais conjuntos serão utilizados na bateria de 5 experimentos realizados para testar a eficácia das heurísticas de sintonização de parâmetros de algoritmos.

O objetivo dos 5 experimentos realizados na etapa seguinte é extrair e comparar dados sobre a eficiência de cada uma das 6 heurísticas de sintonização de parâmetros de algoritmos, com distintos orçamentos de avaliação da função de utilidade. O algoritmo de otimização cujos parâmetros serão sintonizados, nesses experimentos, é um algoritmo da classe da Evolução Diferencial (DE), com quatro parâmetros que são: $de_1 \in \{4; \dots; 200\}$, o tamanho da população de soluções; $de_2 \in [0; 2]$, o fator de mutação; $de_3 \in [0,1; 0,9]$, a taxa de cruzamento; e $de_4 \in \{\text{Rand}/1/\text{Bin}, \text{Best}/1/\text{Bin}, \text{Mean}/1/\text{Bin}, \text{Rand}/1/\text{Exp}, \text{Best}/1/\text{Exp}, \text{Mean}/1/\text{Exp}\}$, a estratégia de mutação adotada. No caso do parâmetro discreto de_4 , os valores foram mapeados de 0 a 5. Basicamente, a DE inicializa a população de soluções com de_1 vetores na região de busca e depois os avalia de acordo com a função-objetivo. No laço de repetição, que dura até findar o total de avaliações, ocorrem as seguintes ações: (i) Seleciona-se a estratégia de_4 ; (ii) Gera-se uma nova população usando de_2 e de_3 ; e (iii) Determina-se a próxima população de pais. Tais ações ocorrem até findar o orçamento

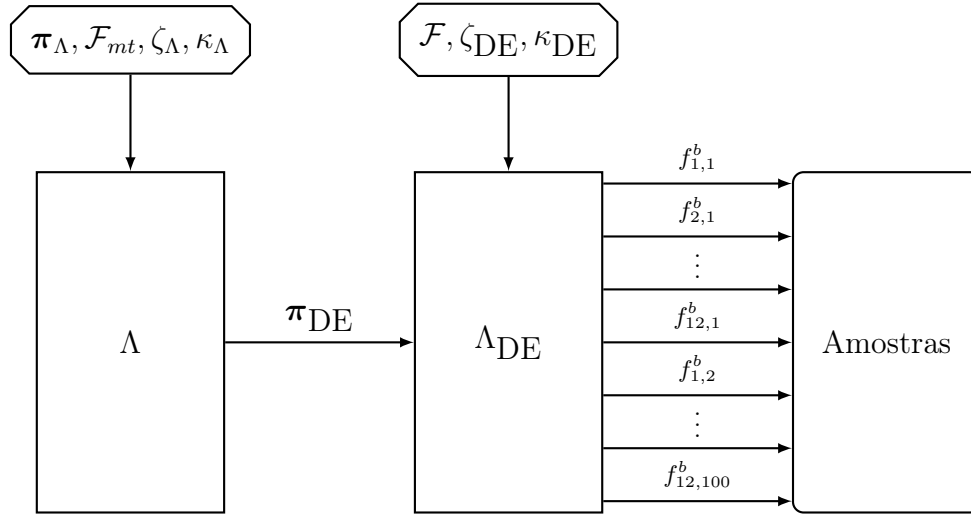


Figura A.2: Processo de extração da amostra de melhores objetivos induzido pelo parâmetro π_{DE} , obtido pela heurística \mathcal{A}_{mt} , e 100 execuções da DE para cada função-objetivo.

de objetivos a serem computados.

Para um experimento qualquer, a Figura A.2 esquematiza o processo de extração das amostras de melhores objetivos induzido por uma única sintonização da DE por uma heurística. Dada uma instância Λ com entradas π_Λ , \mathcal{F}_{mt} , ζ_Λ e κ_Λ , determina-se o conjunto de parâmetros π_{DE} para a DE. Desta forma, com a instância da DE Λ_{DE} com parâmetros fixos π_{DE} , otimiza-se cada um dos objetivos de \mathcal{F} , 100 vezes, com κ_{DE} avaliações, considerando diferentes sementes ζ_{DE} para o gerador de números aleatórios. Ou seja, um π_{DE} gera uma amostra com os melhores 1200 objetivos retornados f_{ij} , para cada função-objetivo $i \in \{1 - \text{Elliptic}, \dots, 12 - \text{HGBat}\}$ e execução $j \in \{1, \dots, 100\}$. Para cada heurística de sintonização este processo é repetido 50 vezes e, portanto, um conjunto de 60000 amostras com melhores objetivos, retornados pelas respectivas DE ajustadas, a representará no processo de análise comparativa.

Foram realizados 5 experimentos, com orçamentos de avaliações de função utilidade de 500, 2000, 3000, 5000 e 10000 avaliações, respectivamente. Foi testada a hipótese nula, com confiança de 0,95, que afirma a igualdade das medianas de melhores objetivos retornados pela DE, induzidas pela aplicação das heurísticas RS, ParamILS, mGA, Revac, bNM e RPS. Tal verificação é realizada pelo teste de Quade com o procedimento de Holm que ajusta o nível de significância na comparação pareada [Conover, 1999; García et al., 2010].

A codificação desta etapa do projeto foi elaborada em C++11 [Stroustrup, 2013] com as bibliotecas Eigen 3 [Guennebaud et al., 2010], GNU Scientific Library [Gough, 2009] e ALGLIB [Bochkanov, 2016], e compilador g++ 5.4.1 com otimização O3. Os

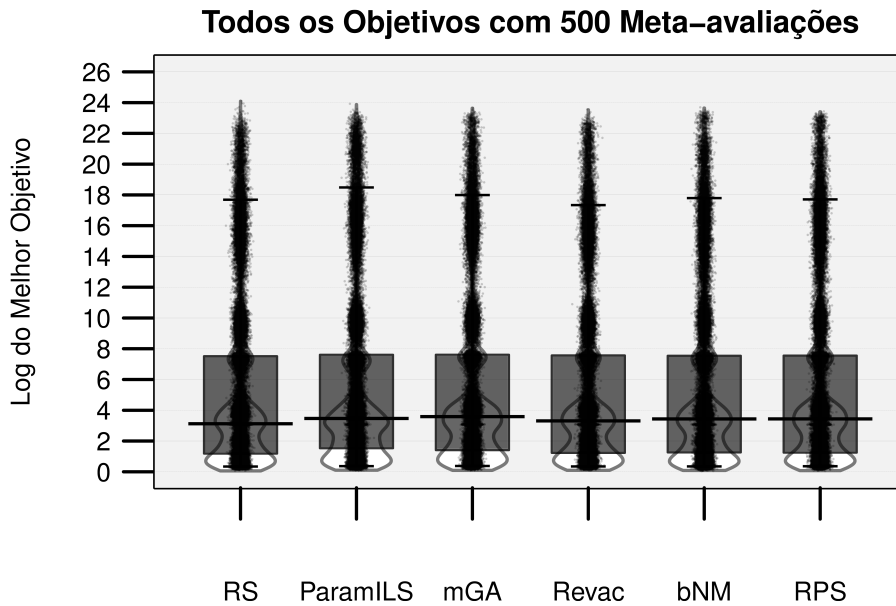


Figura A.3: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.

gráficos e análise dos dados foram gerados no R-project [Matloff, 2011]. Os experimentos foram executados em um computador com processador Intel Core I7, 16 GB de RAM, VGA 4 GB DDR 3 GFORCE GT 630 no Ubuntu 16.04, sem Internet e dedicado aos testes. Foram necessárias aproximadamente duas semanas para completar os testes descritos a seguir. Isto indica o alto custo computacional de problemas desta natureza.

Como resultados, para cada experimento apresentam-se o beanplot das amostras de melhores objetivos retornados e tabelas com estatísticas usuais, além daquela com o resultado do teste estatístico de Quade. Ao final, apresenta-se um gráfico com a evolução da mediana de melhor objetivo com o crescimento no número de avaliações da função de utilidade. Para o experimento com 500 avaliações são apresentados os resultados indicados nas Figura A.3 e Tabelas A.2 e A.3. Para o experimento com orçamento de 2000 avaliações da função de utilidade apresentam-se a Figura A.4 e as Tabelas A.4 e A.5. Para o experimento com orçamento de 3000 avaliações de MBF apresentam-se a Figura A.5 e as Tabelas A.6 e A.7. Para o experimento com orçamento de 5000 avaliações de função utilidade, apresentam-se a Figura A.6 e as Tabelas A.8 e A.9. Para o último experimento, com orçamento de 10000 avaliações, apresentam-se a Figura A.7 e as Tabelas A.10 e A.11. A Figura A.8 ilustra como se comporta a mediana amostral de melhor objetivo, retornado pela DE, pelo número de avaliações da função utilidade pela heurística que fez a sintonia de cada conjunto de parâmetros.

Tabela A.2: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.

Estatística	RS	ParamILS	mGA	Revac	bNM	RPS
Mínimo	0,0525	0,0542	0,0859	0,0664	0,0391	0,0469
1º Quartil	1,1862	1,5352	1,4102	1,2333	1,2640	1,2499
Mediana	3,1219	3,4668	3,5863	3,3082	3,4358	3,4389
3º Quartil	7,5122	7,6058	7,6085	7,5649	7,5419	7,5515
Máximo	24,1011	23,9249	23,6441	23,5626	23,6939	23,4321

Tabela A.3: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 500 avaliações.

	RS	ParamILS	mGA	Revac	bNM
ParamILS	0,000000	–	–	–	–
mGA	0,000000	0,284832	–	–	–
Revac	0,001182	0,000000	0,000000	–	–
bNM	0,000058	0,000000	0,000000	0,453717	–
RPS	0,000000	0,000000	0,000000	0,005450	0,042473
Valor-p do Teste: 0,000000					

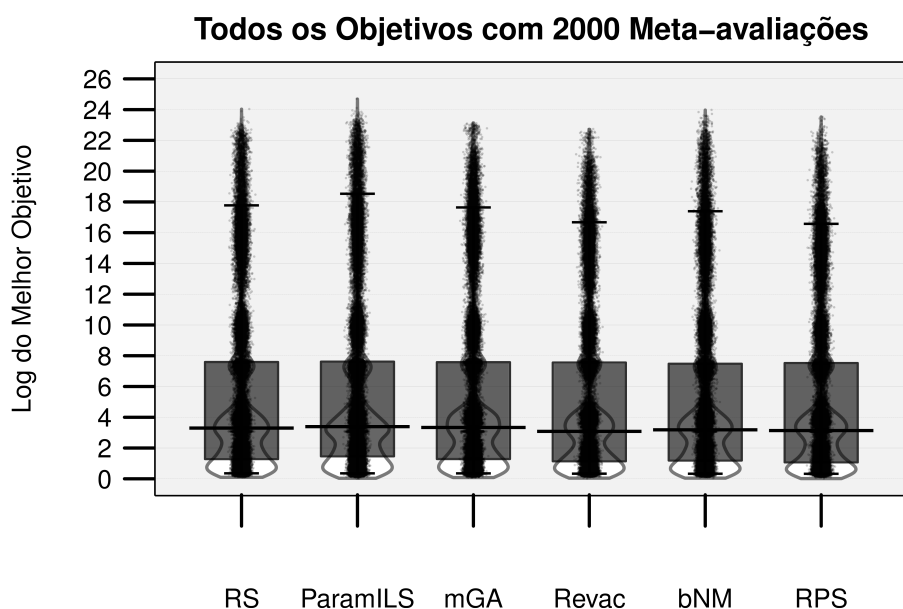


Figura A.4: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.

Tabela A.4: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.

Estatística	RS	ParamILS	mGA	Revac	bNM	RPS
Mínimo	0,0895	0,0101	0,0690	0,0233	0,0306	0,0072
1° Quartil	1,2841	1,4643	1,2869	1,1411	1,1906	1,0713
Mediana	3,2985	3,3912	3,3387	3,0846	3,1852	3,1374
3° Quartil	7,5921	7,6143	7,5811	7,5574	7,4790	7,5235
Máximo	24,0375	24,7112	23,1487	22,7412	23,9821	23,5301

Tabela A.5: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 2000 avaliações.

	RS	ParamILS	mGA	Revac	bNM
ParamILS	0,000000	–	–	–	–
mGA	0,583943	0,000000	–	–	–
Revac	0,000000	0,000000	0,000000	–	–
bNM	0,000000	0,000000	0,000000	0,000833	–
RPS	0,000000	0,000000	0,000000	0,000042	0,000000
Valor-p do Teste: 0,000000					

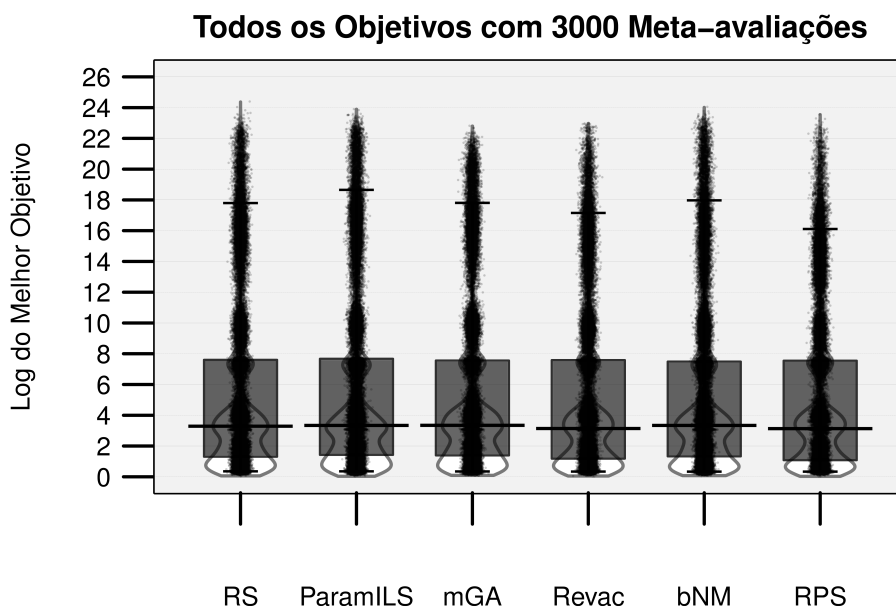


Figura A.5: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.

Tabela A.6: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.

Estatística	RS	ParamILS	mGA	Revac	bNM	RPS
Mínimo	0,0302	0,0186	0,0730	0,0134	0,0364	0,0221
1º Quartil	1,3092	1,4283	1,3925	1,1824	1,3301	1,0815
Mediana	3,2889	3,3413	3,3472	3,1385	3,3410	3,1318
3º Quartil	7,6016	7,6704	7,5564	7,5869	7,4845	7,5434
Máximo	24,3977	23,9069	22,8206	23,0040	24,0531	23,5500

Tabela A.7: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 3000 avaliações.

	RS	ParamILS	mGA	Revac	bNM
ParamILS	0,000000	–	–	–	–
mGA	0,001178	0,000000	–	–	–
Revac	0,000000	0,000000	0,000000	–	–
bNM	0,000000	0,000000	0,000000	0,001178	–
RPS	0,000000	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000					

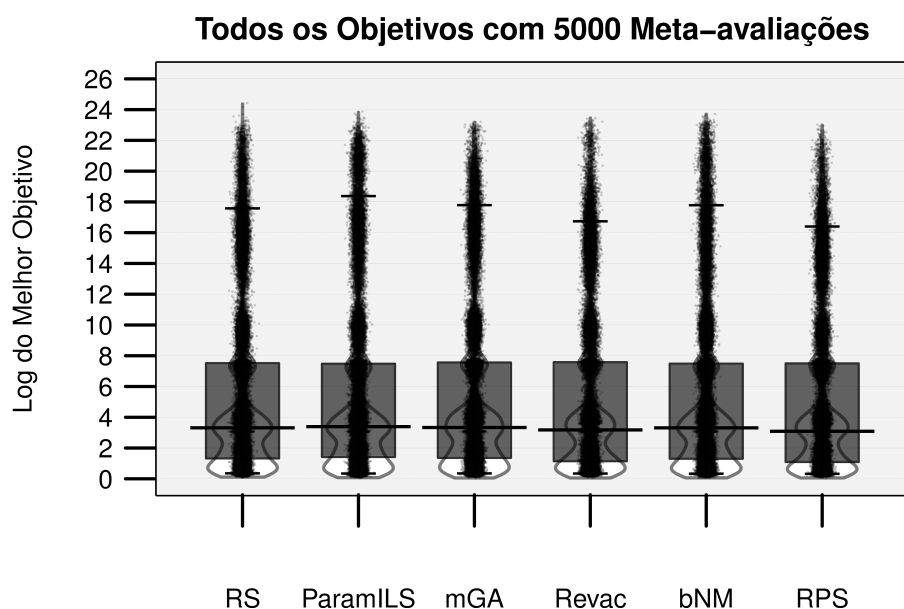


Figura A.6: Beanplot para o $\log(f(x) + 1)$ dos valores de objetivo objetivos retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.

Tabela A.8: Estatísticas para o $\log(f(x) + 1)$ dos valores de objetivo objetivos retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.

Estatística	RS	ParamILS	mGA	Revac	bNM	RPS
Mínimo	0,0618	0,0218	0,0137	0,0486	0,0436	0,0180
1° Quartil	1,3280	1,4207	1,3615	1,1435	1,3059	1,0936
Mediana	3,3138	3,3927	3,3336	3,1772	3,3115	3,0877
3° Quartil	7,5153	7,4779	7,5594	7,5773	7,4838	7,4995
Máximo	24,4259	23,8751	23,2131	23,4544	23,7194	23,0285

Tabela A.9: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 5000 avaliações.

	RS	ParamILS	mGA	Revac	bNM
ParamILS	0,000000	–	–	–	–
mGA	0,000000	0,000427	–	–	–
Revac	0,000427	0,000000	0,000000	–	–
bNM	0,053744	0,000004	0,000000	0,000000	–
RPS	0,000000	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000					

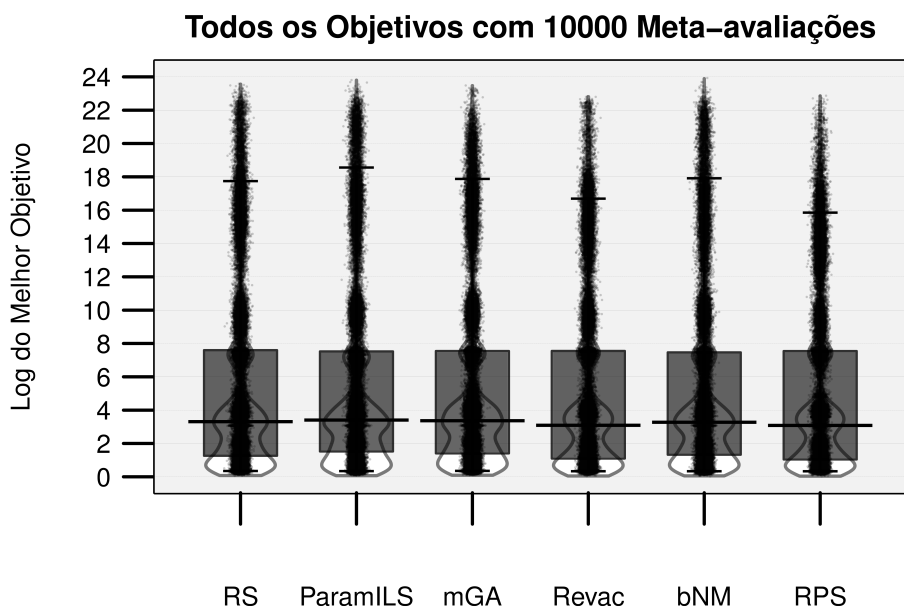


Figura A.7: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.

Tabela A.10: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.

Estatística	RS	ParamILS	mGA	Revac	bNM	RPS
Mínimo	0,0459	0,0622	0,0821	0,0336	0,0183	0,0286
1º Quartil	1,2617	1,5212	1,4059	1,0945	1,3245	1,0404
Mediana	3,3073	3,4053	3,3657	3,0904	3,2741	3,0843
3º Quartil	7,5976	7,5244	7,5462	7,5509	7,4653	7,5444
Máximo	23,5934	23,8075	23,4828	22,8276	23,9309	22,8703

Tabela A.11: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação aos melhores valores de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros com 10000 avaliações.

	RS	ParamILS	mGA	Revac	bNM
ParamILS	0,000000	–	–	–	–
mGA	0,000000	0,000336	–	–	–
Revac	0,000000	0,000000	0,000000	–	–
bNM	0,000241	0,000000	0,000000	0,000000	–
RPS	0,000000	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000					

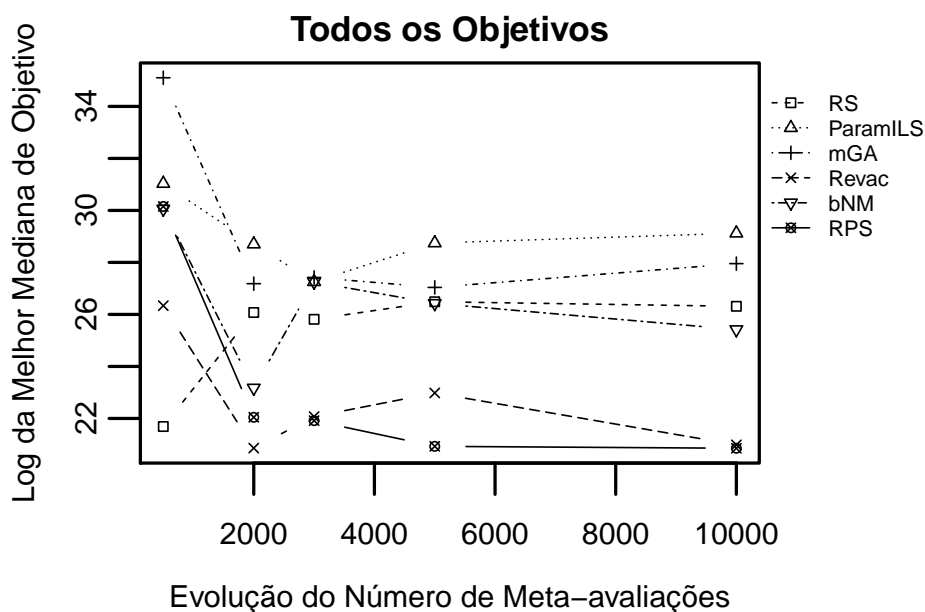


Figura A.8: Evolução das medianas do $\log(f(x) + 1)$ para o melhor valor de objetivo retornados pela DE, induzidos pelos conjuntos de parâmetros ajustados pelas heurísticas de sintonização de parâmetros para 500, 2000, 3000, 5000 e 10000 avaliações.

De um modo geral, esses resultados sugerem as seguintes conclusões:

- Para 500 avaliações da função de utilidade, a busca aleatória RS parece ser mais vantajosa, sendo recomendada para quando se dispõe de pequeno orçamento de avaliações destinadas à sintonização de parâmetros. Isso pode ser observado nas três quatro primeiras estatísticas extraídas e no teste de Quade. Uma possível explicação para isso é que, para tão pequeno orçamento de avaliações de função utilidade, as demais heurísticas teriam feito um investimento na extração de informações do problema que não seria recompensado, uma vez que não teria havido tempo para utilizar a informação adquirida.
- Para 2000 avaliações da função de utilidade, o Revac obteve melhor resultado em relação a mediana, conforme se pode observar no beanplot e nas tabelas com as estatísticas e teste de Quade. Por outro lado, a heurística de sintonização proposta, RPS, obteve um mínimo e primeiro quartil consideravelmente melhores do que todas as demais heurísticas.
- Para orçamentos de 3000 até 10000 avaliações de função de utilidade, o RPS assumiu a melhor colocação no que diz respeito à mediana, de maneira significativa, em relação às demais heurísticas de sintonização.

A.5 Definição dos Parâmetros dos Algoritmos

Finalmente, tendo sido estabelecida a adequação do uso da heurística de sintonização de parâmetros RPS, aqui proposta, esta é empregada para a definição dos parâmetros a serem utilizados nos algoritmos ES-BS, ES-QN, ES-AP, ES-CC e ES-APCC. São necessárias duas sintonizações distintas, uma vez que esses algoritmos serão empregados em dois contextos diferentes: funções não ruidosas e funções ruidosas.

Dois procedimentos de sintonização foram então realizados. O primeiro, para determinar os parâmetros a serem utilizados pelas três primeiras estratégias (ES-BS, ES-QN e ES-AP) em um ambiente sem ruído na função-objetivo. O segundo para determinar os parâmetros a serem utilizados pelos cinco algoritmos em ambientes com funcionais ruidosos.

Nos dois procedimentos, os valores dos sete parâmetros para o RPS foram escolhidos de forma *ad hoc*, ou seja, da forma como é realizada nos trabalhos sobre heurísticas de sintonização citados no trabalho [Eiben & Smit, 2011]. Assim sendo, foram utilizados $\delta_e = 1,2$, $\delta_c = 0,9$, $\delta_s = 0,8$, $f_p = 0,05$, $\varepsilon = 0,01$, $e_{\max} = 3$ e $p_\alpha = 0,9$ em todos os procedimentos relatados nesta seção.

Em ambos os procedimentos, executou-se o RPS 20 vezes para cada algoritmo, sendo feita a minimização da função de utilidade, considerando as funções-objetivo Elliptic, Rosenbrock e Rastrigin, que foram avaliadas 500 vezes pelo algoritmo a ser sintonizado, com limites entre -100 e 100 para as variáveis, na dimensão 10. Tal escolha de funções para comporem o conjunto \mathcal{F} no processo de sintonização de parâmetros se justifica pela diversidade de características das mesmas, visando a adaptação dos algoritmos a serem sintonizados para uma aptidão generalista. Para o treinamento ruidoso, adotou-se o valor de desvio igual a $0,01$. Em cada execução do RPS foram avaliadas exatamente 300 funções de utilidade.

Para cada estratégia evolutiva, os 20 candidatos a parâmetros foram ordenados de forma crescente pelos valores da função utilidade, sendo os 5 primeiros selecionados para uma etapa final. Nesta etapa, para cada um dos cinco candidatos, extraiu-se uma amostra de 100 valores de função utilidade. Essas cinco amostras foram analisadas pela mediana utilizando o teste de Quade com comparações pareadas em um nível de confiança de 95%. Em caso de empate no primeiro lugar, foi utilizada como critério de desempate a menor mediana amostral encontrada.

Os resultados para cada estratégia evolutiva considerada, no contexto de função-objetivo não ruidosa, são apresentados nos seguintes itens:

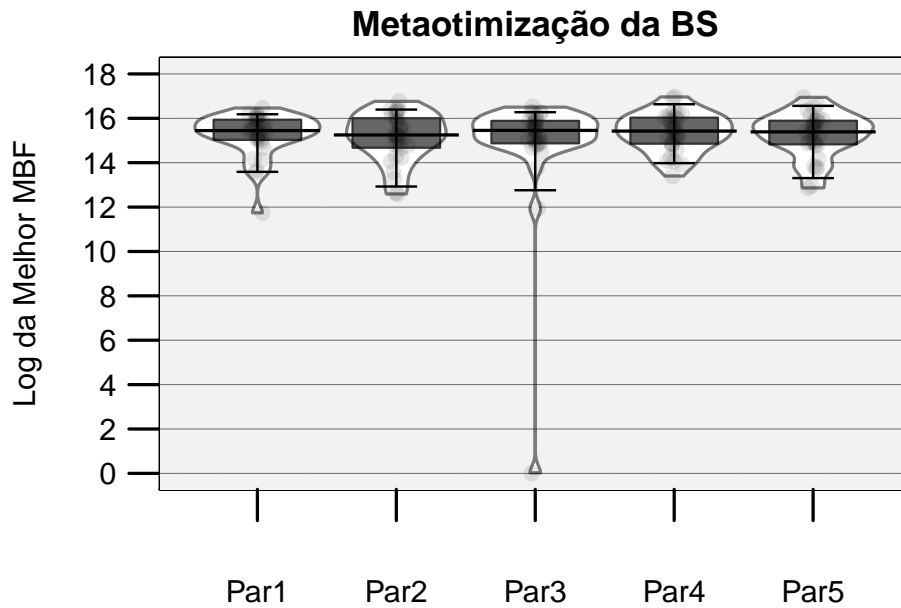
- ES-BS: Tabela A.12, Figura A.9, Tabela A.13 e Tabela A.14. Conjunto de parâmetros selecionado Par2: $(p_u, p_s, p_c) = (0,856; 9,035; 0,674)$.
- ES-QN: Tabela A.15, Figura A.10, Tabela A.13 e Tabela A.17. Conjunto de parâmetros selecionado Par3: $(p_u, p_s, p_c, p_l, p_\varepsilon) = (1,038; 14,506; 0,712; 0,129; 0,068)$.
- ES-AP: Tabela A.18, Figura A.11, Tabela A.19 e Tabela A.20. Conjunto de parâmetros selecionado Par1: $(p_u, p_s, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}) = (0,885; 14,080; 0,888; 0,271; 0,009; 2,511; 1,475)$.

Os resultados para cada estratégia evolutiva, para o caso de funções ruidosas, são apresentados nos seguintes itens:

- ES-BS: Tabela A.21, Figura A.12, Tabela A.22 e Tabela A.23. Conjunto de parâmetros selecionado Par1: $(p_u, p_s, p_c) = (0,926; 7,531; 0,757)$.
- ES-QN: Tabela A.24, Figura A.13, Tabela A.22 e Tabela A.26. Conjunto de parâmetros selecionado Par1: $(p_u, p_s, p_c, p_l, p_\varepsilon) = (0,841; 12,860; 0,505; 0,444; 0,345)$.

Tabela A.12: Candidatos a conjunto de parâmetros para o algoritmo ES-BS em objetivos não ruidosos.

Conjunto	Parâmetros		
	p_u	p_s	p_c
Par1	0,858	10,973	0,828
Par2	0,856	9,035	0,674
Par3	1,077	8,580	0,758
Par4	1,162	12,865	0,795
Par5	0,944	13,973	0,856

Figura A.9: Beanplot para o $\log(\nu+1)$ dos melhores valores de MBF, relativo ao mínimo 231859,8154, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos não ruidosos.

- ES-AP: Tabela A.27, Figura A.14, Tabela A.28 e Tabela A.29. Conjunto de parâmetros selecionado Par4: $(p_u, p_s, p_c, p_l, p_\varepsilon, p_{db}, p_{sp}) = (1,143; 14,508; 0,659; 0,114; 0,009; 2,973; 1,817)$.
- ES-CC: Tabela A.30, Figura A.15, Tabela A.31 e Tabela A.32. Conjunto de parâmetros selecionado Par4: $(p_u, p_s, p_c, p_E, p_\alpha) = (0,805; 6,231; 0,899; 2; 0,727)$.
- ES-APCC: Tabela A.33, Figura A.16, Tabela A.34 e Tabela A.35. Conjunto de parâmetros selecionado Par1: $(p_u, p_s, p_c, p_E, p_\alpha, p_l, p_\varepsilon, p_{db}, p_{sp}) = (0,806; 5,786; 0,563; 2; 0,856; 0,368; 0,338; 2,967; 2,033)$.

Tabela A.13: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 231859,8154, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos não ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	11,7387	12,5824	0,0000	13,3913	12,8571
1° Quartil	15,0301	14,6765	14,8727	14,8571	14,8288
Mediana	15,4460	15,2503	15,4552	15,4252	15,3894
3° Quartil	15,9365	16,0054	15,8851	16,0277	15,8906
Máximo	16,4658	16,7713	16,5244	16,9706	16,9511

Tabela A.14: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-BS em objetivos não ruidosos.

	Par1	Par2	Par3	Par4
Par2	1,000000	–	–	–
Par3	1,000000	1,000000	–	–
Par4	1,000000	1,000000	1,000000	–
Par5	1,000000	1,000000	1,000000	1,000000
Valor-p do Teste: 0,005842				

Tabela A.15: Candidatos a conjunto de parâmetros para o algoritmo ES-QN em objetivos não ruidosos.

Conjunto	Parâmetros				
	p_u	p_s	p_c	p_l	p_ε
Par1	1,020	7,085	0,938	0,038	0,066
Par2	0,870	10,042	0,978	0,030	0,218
Par3	1,038	14,506	0,712	0,129	0,068
Par4	1,190	5,154	0,576	0,402	0,128
Par5	1,117	5,179	0,741	0,157	0,326

Tabela A.16: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 8549,8055, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos não ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	0,0000	10,4213	6,3521	9,3816	11,4480
1° Quartil	12,3651	12,9815	12,2706	12,6101	13,6048
Mediana	13,5694	14,4477	13,3916	13,7524	14,3398
3° Quartil	14,5458	15,3978	14,8771	14,8196	15,1833
Máximo	16,0592	16,6106	16,2410	17,7168	17,0967

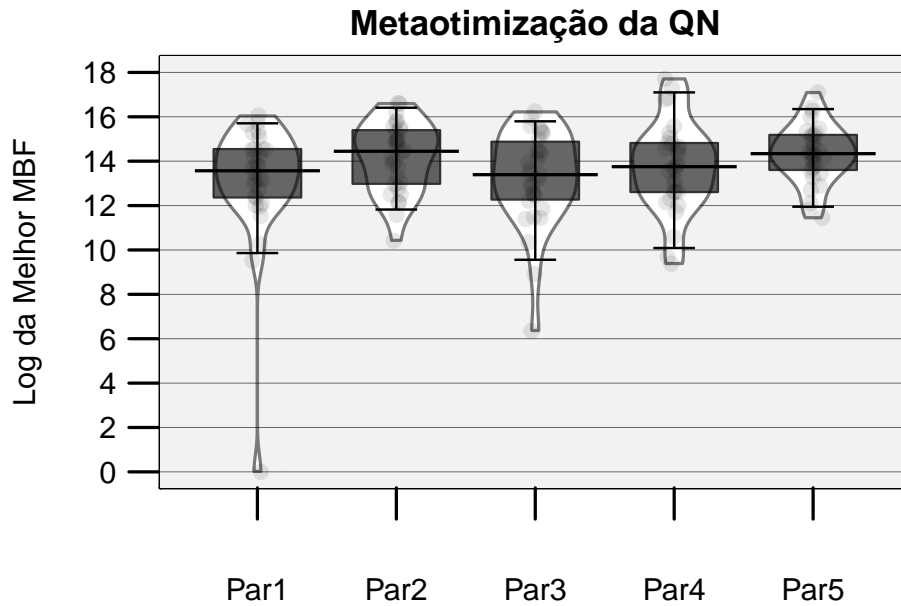


Figura A.10: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 8549,8055, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos não ruidosos.

Tabela A.17: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-QN em objetivos não ruidosos.

	Par1	Par2	Par3	Par4
Par2	0,942959	–	–	–
Par3	1,000000	0,599370	–	–
Par4	1,000000	1,000000	1,000000	–
Par5	1,000000	1,000000	0,673151	1,000000
Valor-p do Teste: 0,221340				

Tabela A.18: Candidatos a conjunto de parâmetros para o algoritmo ES-AP em objetivos não ruidosos.

Conjunto	Parâmetros						
	p_u	p_s	p_c	p_l	p_ε	p_{db}	p_{sp}
Par1	0,885	14,080	0,888	0,271	0,009	2,511	1,475
Par2	1,040	9,473	0,771	0,341	0,001	1,353	1,381
Par3	0,831	5,043	0,890	0,305	0,091	3,878	2,934
Par4	1,168	5,489	0,762	0,365	0,189	1,359	2,712
Par5	0,886	7,834	0,919	0,381	0,203	2,500	1,679

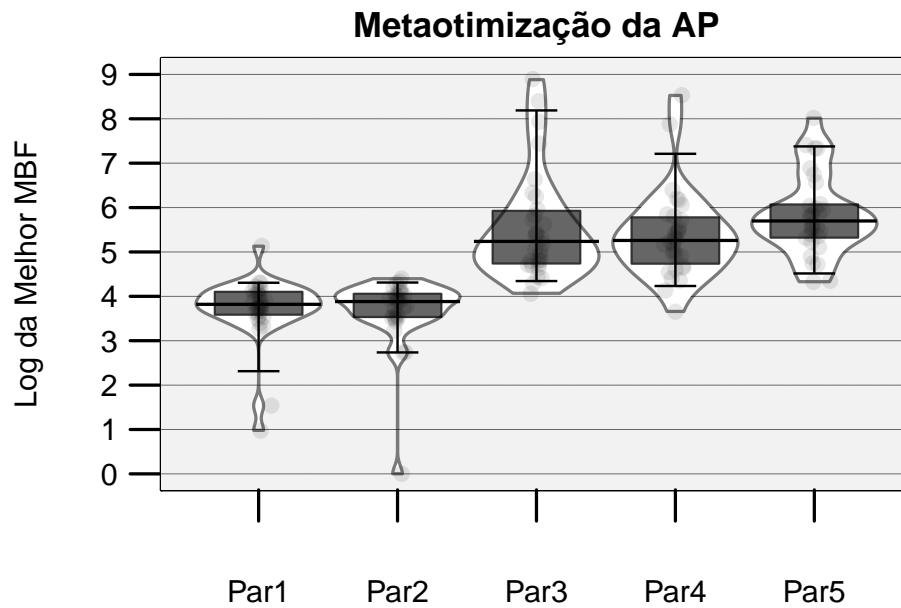


Figura A.11: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 21,3743, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos não ruidosos.

Tabela A.19: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 21,3743, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos não ruidosos.

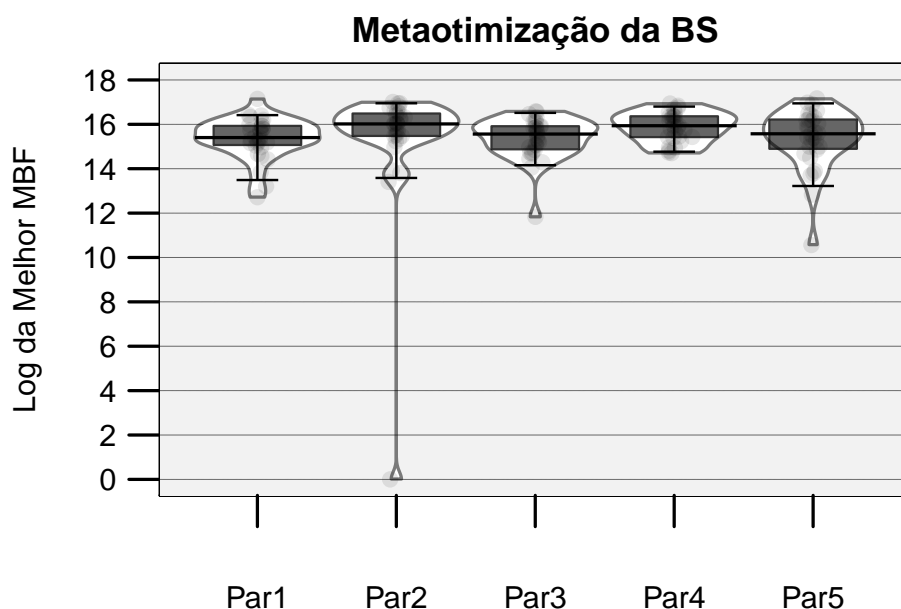
Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	0,9741	0,0000	4,0568	3,6531	4,3258
1º Quartil	3,5872	3,5345	4,7407	4,7369	5,3238
Mediana	3,8198	3,8829	5,2373	5,2592	5,6980
3º Quartil	4,1005	4,0569	5,9253	5,7783	6,0695
Máximo	5,1361	4,3990	8,8945	8,5327	8,0185

Tabela A.20: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-AP em objetivos não ruidosos.

	Par1	Par2	Par3	Par4
Par2	1,000000	–	–	–
Par3	0,000000	0,000000	–	–
Par4	0,000000	0,000000	1,000000	–
Par5	0,000000	0,000000	1,000000	1,000000
Valor-p do Teste: 0,000000				

Tabela A.21: Candidatos a conjunto de parâmetros para o algoritmo ES-BS em objetivos ruidosos.

Conjunto	Parâmetros		
	p_u	p_s	p_c
Par1	0,926	7,531	0,757
Par2	0,862	5,483	0,537
Par3	0,915	5,890	0,695
Par4	1,162	11,476	0,637
Par5	1,023	12,359	0,598

Figura A.12: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 252141,1039, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos ruidosos.Tabela A.22: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 252141,1039, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-BS em objetivos ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	12,7155	0,0000	11,8311	14,7039	10,5609
1º Quartil	15,0696	15,4777	14,8747	15,4257	14,8967
Mediana	15,4047	16,0180	15,5620	15,9381	15,5748
3º Quartil	15,9296	16,4866	15,9139	16,3627	16,2164
Máximo	17,1472	17,0087	16,5846	16,9370	17,1666

Tabela A.23: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-BS em objetivos ruidosos.

	Par1	Par2	Par3	Par4
Par2	0,714086	–	–	–
Par3	1,000000	0,982105	–	–
Par4	0,346626	1,000000	0,634581	–
Par5	1,000000	1,000000	1,000000	0,720451
Valor-p do Teste: 0,142775				

Tabela A.24: Candidatos a conjunto de parâmetros para o algoritmo ES-QN em objetivos ruidosos.

Conjunto	Parâmetros				
	p_u	p_s	p_c	p_l	p_ε
Par1	0,841	12,860	0,505	0,444	0,345
Par2	1,075	9,827	0,525	0,357	0,188
Par3	0,822	7,024	0,721	0,258	0,151
Par4	1,171	6,117	0,548	0,216	0,405
Par5	1,184	13,309	0,665	0,447	0,484

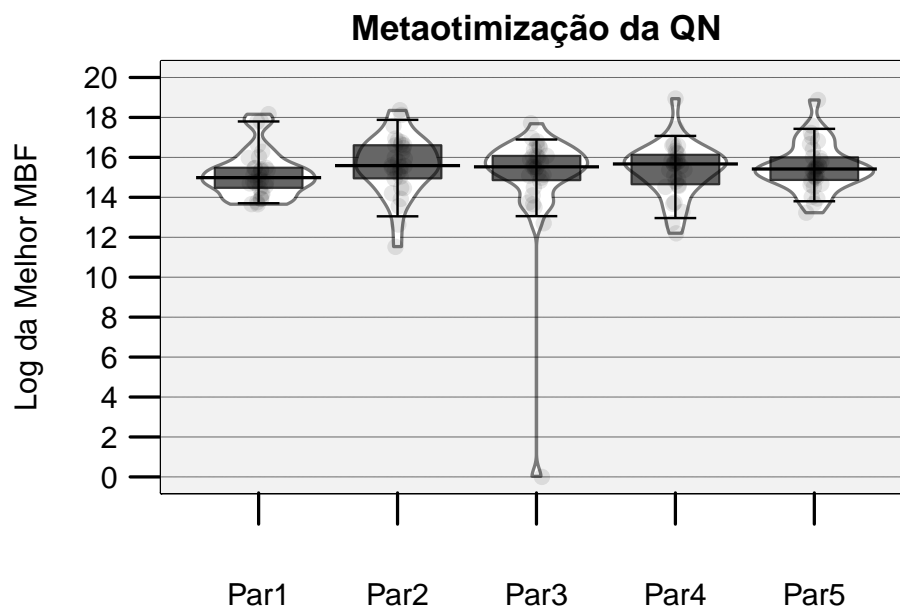


Figura A.13: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 366736,8967, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos ruidosos.

Tabela A.25: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 366736,8967, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-QN em objetivos ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	13,6489	11,5183	0,0000	12,1959	13,2279
1° Quartil	14,4786	14,9527	14,8602	14,6558	14,8716
Mediana	14,9858	15,5870	15,5263	15,6686	15,4179
3° Quartil	15,4787	16,6047	16,0706	16,1263	15,9978
Máximo	18,1648	18,3636	17,7126	18,9462	18,8742

Tabela A.26: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-QN em objetivos ruidosos.

	Par1	Par2	Par3	Par4
Par2	1,000000	–	–	–
Par3	1,000000	1,000000	–	–
Par4	1,000000	1,000000	1,000000	–
Par5	1,000000	1,000000	1,000000	1,000000
Valor-p do Teste: 0,985973				

Tabela A.27: Candidatos a conjunto de parâmetros para o algoritmo ES-AP em objetivos ruidosos.

Conjunto	Parâmetros						
	p_u	p_s	p_c	p_l	p_ε	p_{db}	p_{sp}
Par1	1,158	12,165	0,536	0,421	0,185	2,553	2,539
Par2	1,171	13,658	0,536	0,323	0,592	4,580	1,817
Par3	0,975	10,316	0,527	0,483	0,835	1,881	2,404
Par4	1,143	14,508	0,659	0,114	0,009	2,973	1,817
Par5	1,082	5,468	0,560	0,484	0,064	2,704	1,370

Tabela A.28: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-45864,5972$, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	8,3657	9,4665	9,2276	8,2114	0,0000
1° Quartil	9,7041	9,8523	9,8327	9,7156	9,3630
Mediana	9,8395	9,9901	10,0943	9,8084	9,8563
3° Quartil	9,9902	10,1180	10,2928	10,0210	9,9478
Máximo	10,4329	10,5638	10,6026	10,4540	10,4999

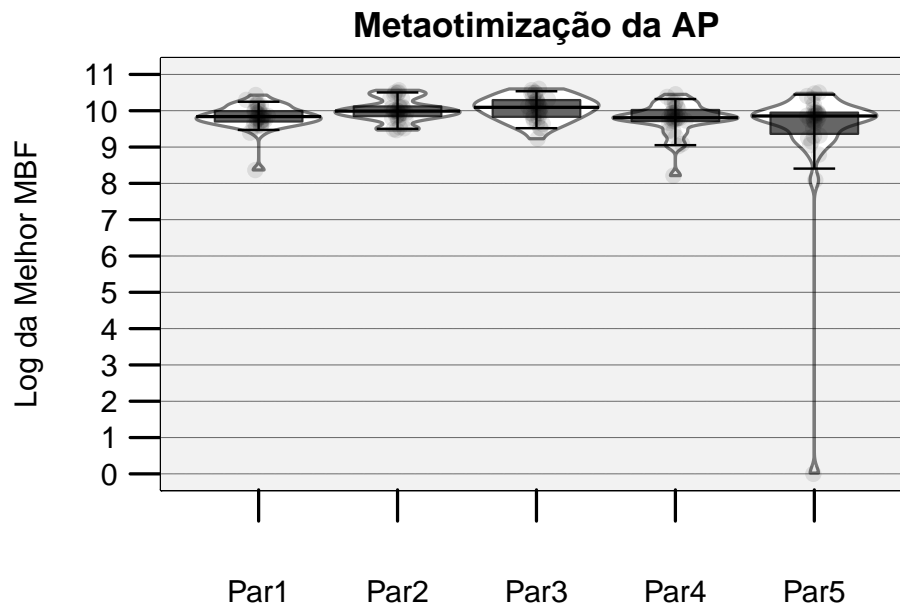


Figura A.14: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-45864,5972$, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-AP em objetivos ruidosos.

Tabela A.29: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-AP em objetivos ruidosos.

	Par1	Par2	Par3	Par4
Par2	0,059889	–	–	–
Par3	0,017824	1,000000	–	–
Par4	1,000000	0,120546	0,038312	–
Par5	1,000000	0,027410	0,005895	1,000000
Valor-p do Teste: 0,000630				

Tabela A.30: Candidatos a conjunto de parâmetros para o algoritmo ES-CC em objetivos ruidosos.

Conjunto	Parâmetros				
	p_u	p_s	p_c	p_E	p_α
Par1	0,936	6,817	0,703	2,492	0,834
Par2	0,921	9,290	0,692	4,976	0,779
Par3	1,050	9,594	0,775	4,831	0,876
Par4	0,805	6,231	0,899	2,546	0,727
Par5	0,969	11,604	0,654	2,101	0,619

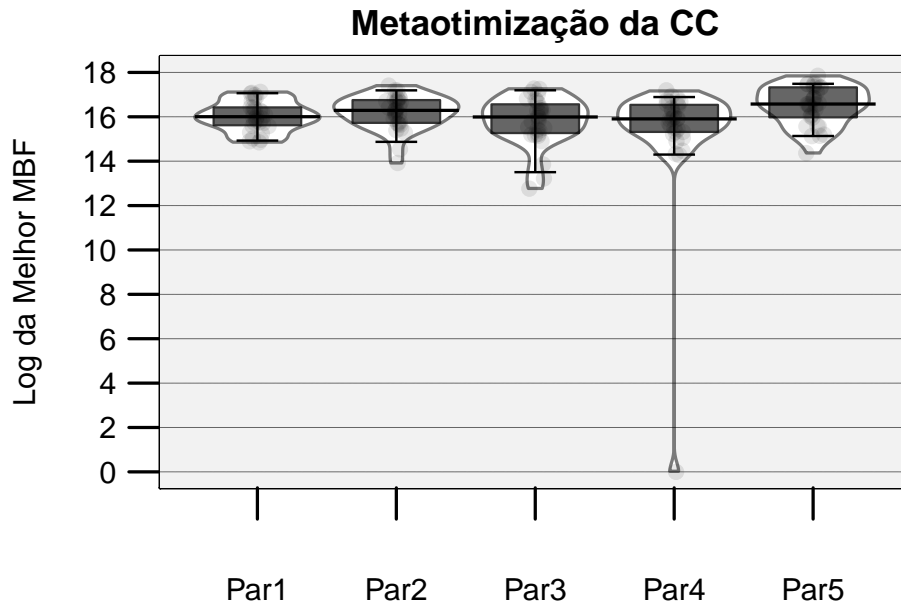


Figura A.15: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 404031,8039, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-CC em objetivos ruidosos.

Tabela A.31: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo 404031,8039, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-CC em objetivos ruidosos.

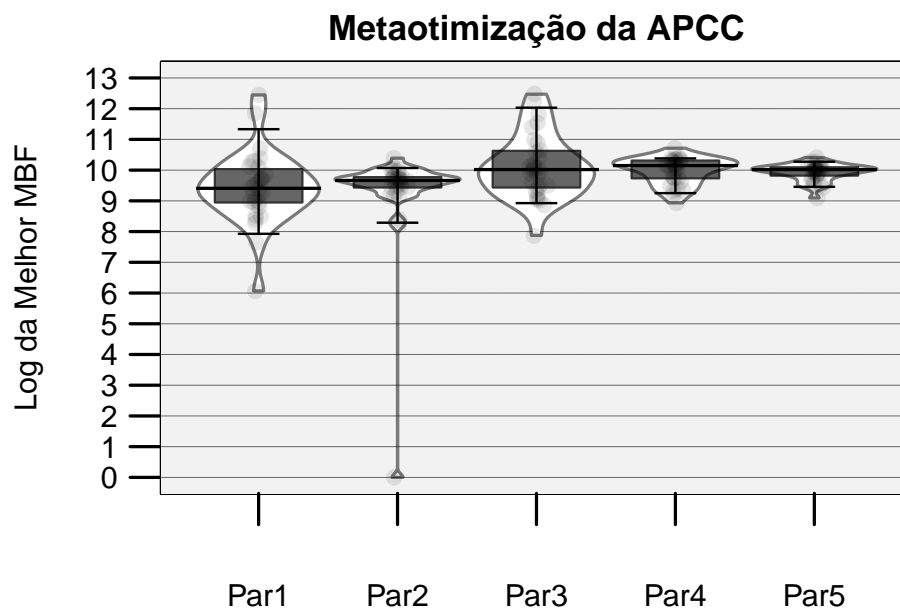
Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	14,8493	13,9195	12,7633	0,0000	14,3651
1º Quartil	15,6199	15,7281	15,2690	15,3186	15,9762
Mediana	16,0049	16,2896	15,9917	15,9069	16,5750
3º Quartil	16,4227	16,7530	16,5618	16,5370	17,3261
Máximo	17,1232	17,4064	17,2712	17,1934	17,8487

Tabela A.32: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-CC em objetivos ruidosos.

	Par1	Par2	Par3	Par4
Par2	1,000000	–	–	–
Par3	1,000000	1,000000	–	–
Par4	1,000000	0,391128	1,000000	–
Par5	0,169523	1,000000	0,176489	0,028247
Valor-p do Teste: 0,023516				

Tabela A.33: Candidatos a conjunto de parâmetros para o algoritmo ES-APCC em objetivos ruidosos.

Conjunto	Parâmetros								
	p_u	p_s	p_c	p_E	p_α	p_l	p_ε	p_{db}	p_{sp}
Par1	0,806	5,786	0,563	2,942	0,856	0,368	0,338	2,967	2,033
Par2	1,013	12,903	0,626	2,980	0,835	0,212	0,177	1,605	2,958
Par3	1,072	5,443	0,530	2,371	0,624	0,407	0,761	1,478	2,456
Par4	0,862	11,697	0,706	3,595	0,823	0,196	0,592	4,641	1,980
Par5	1,065	14,771	0,670	2,644	0,715	0,327	0,411	1,193	2,876

Figura A.16: Beanplot para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-24167,0778$, retornados pelos conjuntos de candidatos a parâmetros do algoritmo ES-APCC em objetivos ruidosos.Tabela A.34: Estatísticas para o $\log(\nu + 1)$ dos melhores valores de MBF, relativo ao mínimo $-24167,0778$, retornadas pelos conjuntos de candidatos a parâmetros do algoritmo ES-APCC em objetivos ruidosos.

Estatística	Par1	Par2	Par3	Par4	Par5
Mínimo	6,0652	0,0000	7,8655	8,9315	9,0908
1º Quartil	8,9477	9,4350	9,4354	9,7345	9,8214
Mediana	9,4087	9,6635	10,0191	10,1478	10,0171
3º Quartil	10,0328	9,7720	10,6276	10,3106	10,0906
Máximo	12,4511	10,3921	12,4824	10,7148	10,4087

Tabela A.35: Valores-p geral e das comparações pareadas entre os valores de MBF dos candidatos a conjunto de parâmetros do algoritmo ES-APCC em objetivos ruidosos.

	Par1	Par2	Par3	Par4
Par2	1,000000	–	–	–
Par3	0,004689	0,005355	–	–
Par4	0,013349	0,014555	1,000000	–
Par5	0,241186	0,241186	0,519625	0,764968
Valor-p do Teste: 0,000280				

Apêndice B

Resultados Extras

B.1 Apresentação

Apresenta-se aqui uma miscelânea de figuras e tabelas referentes aos resultados extras deste trabalho. Gráficos beanplot e da mediana do melhor objetivo pela evolução das avaliações, juntamente com tabelas de estatísticas e do teste de Quade são apresentados para cada função-objetivo no experimento sem ruído na avaliação de objetivo; para cada função-objetivo e todos os níveis de ruído, e para cada função-objetivo com cada nível de ruído para o experimento com objetivos ruidosos. Para a terceira experimentação, são listados em tabelas o conjunto de taxas de vacinação encontradas pelas quatro estratégias evolutivas em trinta execuções.

B.2 Comparações extras entre os Algoritmos

Para o experimento que compara as estratégias evolutivas ES-BS, ES-QN e ES-AP em um benchmark com 12 funções-objetivo sem ruído, o resultado em separado para cada uma delas pode ser observado nos beanplots da Figura B.1; na Tabela B.1 com as estatísticas de mínimo, primeiro, segundo e terceiro quartis, e máximo; pela Tabela B.2 com os resultados do teste de Quade; e pelos gráficos da Figura B.2 com a evolução da mediana do melhor valor objetivo pelo número de avaliações.

Para o segundo experimento com objetivos ruidosos, em relação as estratégias ES-BS, ES-QN, ES-AP e ES-APCC, considerando cada função, entre 12, em todos os níveis de desvio padrão, a Figura B.3 ilustra os beanplots por função-objetivo; a Tabela B.3 exhibe estatísticas para o logaritmo dos melhores objetivos reais retornados pelas heurísticas; a Tabela B.4 revela os valores-P do teste de Quade para cada função; e a

Tabela B.1: Estatísticas para o $\log(f(x)+1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação à cada função-objetivo no experimento sem ruído.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	13,2983	10,2881	0,0488
1° Quartil	15,5483	12,6258	0,2486
Mediana	15,9245	14,3364	0,4505
3° Quartil	16,5665	16,3159	0,7611
Máximo	17,2193	17,8617	6,3313

(a) Elliptic.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	1,3296	4,8820	0,4390
1° Quartil	7,1037	6,1450	1,0968
Mediana	8,1124	7,1102	1,5228
3° Quartil	8,5320	7,6387	2,1408
Máximo	9,3744	9,0345	2,7620

(b) Bent Cigar.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	9,9959	8,8591	0,0404
1° Quartil	10,8928	10,6224	0,0879
Mediana	11,0897	10,9881	0,1507
3° Quartil	11,3875	11,2879	0,2085
Máximo	11,7559	11,5845	10,9260

(c) Discus.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	0,3867	0,0610	0,1137
1° Quartil	3,2409	3,6256	2,1728
Mediana	3,5934	3,9485	3,5824
3° Quartil	3,6214	4,2768	3,6016
Máximo	4,2224	6,0568	4,2256

(d) Rosenbrock.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	3,0445	3,0458	3,0442
1° Quartil	3,0445	3,0702	3,0445
Mediana	3,0445	3,0751	3,0445
3° Quartil	3,0445	3,0798	3,0445
Máximo	3,0445	3,0917	3,0445

(e) Ackley.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	1,9478	2,3772	2,2307
1° Quartil	2,6480	2,6643	2,6338
Mediana	2,8263	2,7840	2,8009
3° Quartil	2,9324	2,8257	2,9382
Máximo	3,1193	3,0047	3,0743

(f) Weierstrass.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	0,0397	0,2097	0,0000
1° Quartil	0,1140	0,4107	0,0000
Mediana	0,2396	0,5092	0,0000
3° Quartil	0,3422	0,5830	0,0011
Máximo	0,6844	0,8431	0,0176

(g) Griewank.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	4,0204	3,5364	3,9346
1° Quartil	4,6575	4,3612	4,7045
Mediana	5,0056	4,6892	4,8507
3° Quartil	5,2806	5,0385	5,2938
Máximo	5,5164	5,3728	5,6717

(h) Rastrigin.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	6,2937	6,7605	5,3857
1° Quartil	7,1946	7,2408	7,2305
Mediana	7,3561	7,3629	7,4238
3° Quartil	7,5617	7,5607	7,5950
Máximo	7,7742	7,9413	7,8505

(i) Schwefel.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	0,3507	0,6625	0,0987
1° Quartil	0,6301	0,9069	0,4325
Mediana	1,0557	1,1087	0,7234
3° Quartil	1,6349	1,1944	1,2239
Máximo	2,4364	1,4703	1,8806

(j) Katsuura.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	0,1184	0,1815	0,1939
1° Quartil	0,2249	0,2551	0,2715
Mediana	0,3068	0,3033	0,3670
3° Quartil	0,4349	0,3814	0,5399
Máximo	0,7066	0,5482	0,6723

(k) HappyCat.

Estatística	ES-BS	ES-QN	ES-AP
Mínimo	0,1615	0,2657	0,2512
1° Quartil	0,2546	0,3144	0,3030
Mediana	0,2802	0,3714	0,3623
3° Quartil	0,4521	0,4889	0,4449
Máximo	0,7936	0,6906	0,5845

(l) HGBat.

Tabela B.2: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação à cada função-objetivo no experimento sem ruído.

	ES-BS	ES-QN
ES-QN	0,011892	–
ES-AP	0,000000	0,000000
Valor-p do Teste: 0,000000		

(a) Elliptic.

	ES-BS	ES-QN
ES-QN	0,650248	–
ES-AP	0,000000	0,000000
Valor-p do Teste: 0,000000		

(c) Discus.

	ES-BS	ES-QN
ES-QN	0,000000	–
ES-AP	0,920817	0,000000
Valor-p do Teste: 0,000000		

(e) Ackley.

	ES-BS	ES-QN
ES-QN	0,000156	–
ES-AP	0,000000	0,000000
Valor-p do Teste: 0,000000		

(g) Griewank.

	ES-BS	ES-QN
ES-QN	1,000000	–
ES-AP	1,000000	1,000000
Valor-p do Teste: 0,835969		

(i) Schwefel.

	ES-BS	ES-QN
ES-QN	0,275763	–
ES-AP	0,275763	0,024755
Valor-p do Teste: 0,029306		

(k) HappyCat.

	ES-BS	ES-QN
ES-QN	0,000749	–
ES-AP	0,000000	0,000000
Valor-p do Teste: 0,000000		

(b) Bent Cigar.

	ES-BS	ES-QN
ES-QN	0,001909	–
ES-AP	0,751086	0,001048
Valor-p do Teste: 0,000000		

(d) Rosenbrock.

	ES-BS	ES-QN
ES-QN	1,000000	–
ES-AP	1,000000	1,000000
Valor-p do Teste: 0,881501		

(f) Weierstrass.

	ES-BS	ES-QN
ES-QN	0,033470	–
ES-AP	0,540618	0,099024
Valor-p do Teste: 0,029175		

(h) Rastrigin.

	ES-BS	ES-QN
ES-QN	0,792934	–
ES-AP	0,072841	0,089900
Valor-p do Teste: 0,047758		

(j) Katsuura.

	ES-BS	ES-QN
ES-QN	0,490733	–
ES-AP	0,788569	0,788569
Valor-p do Teste: 0,370157		

(l) HGBat.

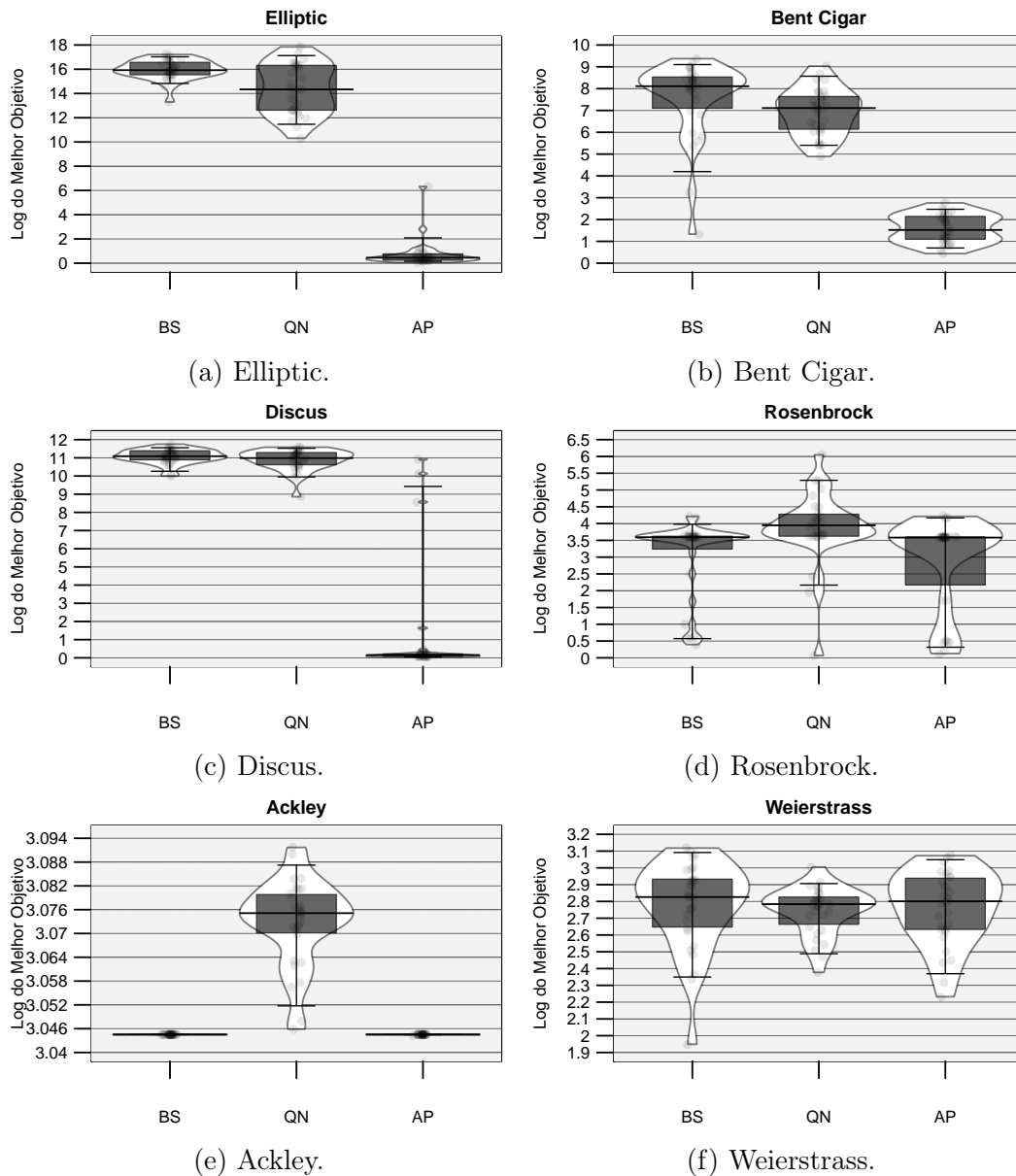


Figura B.4 expõe a evolução da mediana do melhor objetivo real dos algoritmos pelo número de avaliações.

Em relação ao terceiro experimento, as Tabelas B.5, B.6, B.7, B.8 e B.9 apresentam as 30 melhores soluções retornadas, respectivamente, pelas estratégias ES-BS, ES-QN, ES-AP e ES-APCC. Cada linha representa o melhor conjunto de taxas de vacinação encontrado em cada execução.

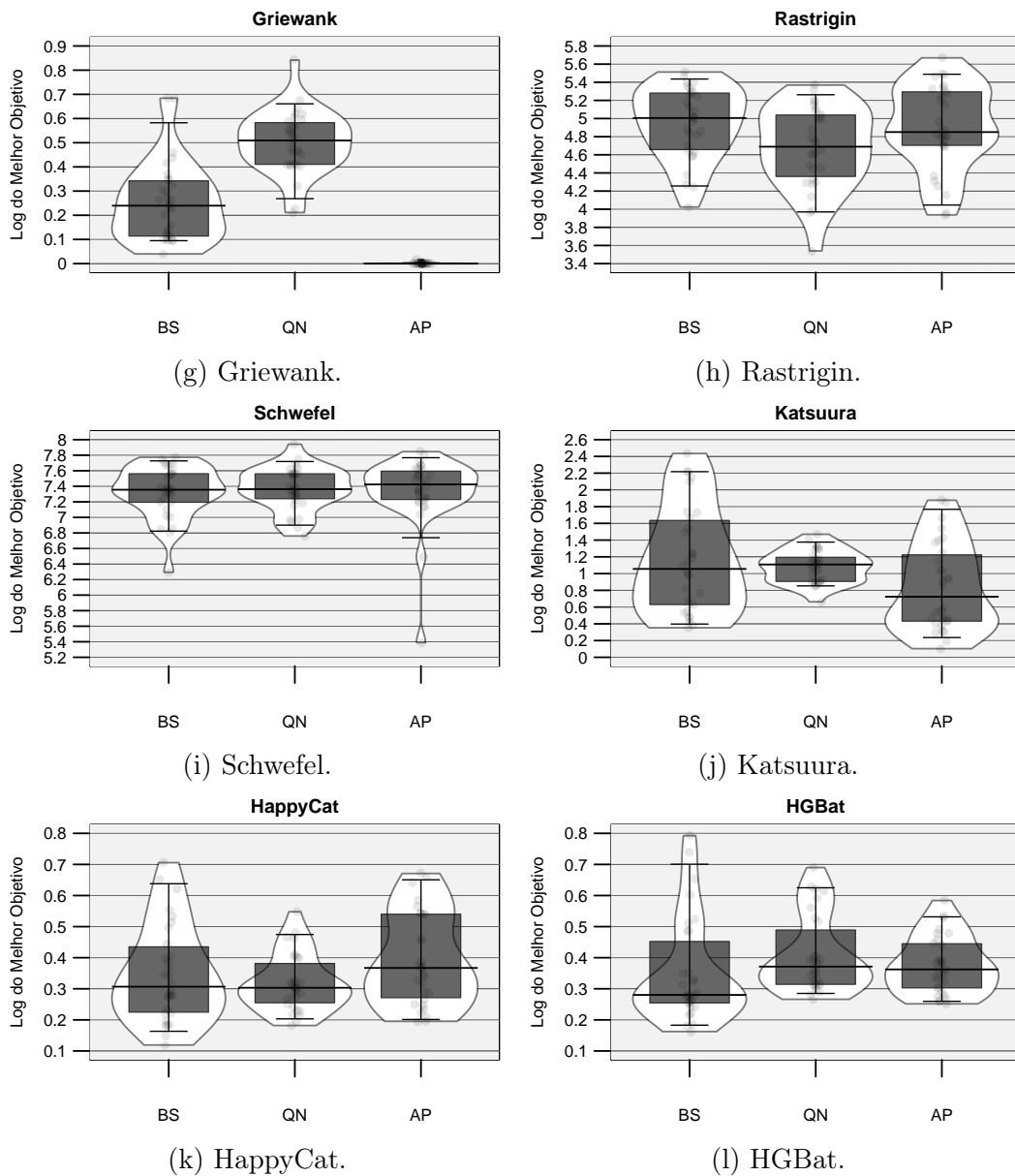


Figura B.1: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo retornados pelos algoritmos em relação à cada função-objetivo no experimento sem ruído.

Tabela B.3: Estatísticas para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.

Estatística	BS	QN	AP	CC	APCC
Mínimo	12,8439	11,7371	0,1890	13,1166	5,5024
1° Quartil	15,4363	14,9943	2,6343	15,8946	8,3519
Mediana	16,0057	15,9714	4,6268	16,4843	9,2713
3° Quartil	16,5511	17,0154	8,0767	17,0643	10,2448
Máximo	18,3034	19,0349	12,8389	18,1859	16,7200

(a) Elliptic.

Estatística	BS	QN	AP	CC	APCC
Mínimo	3,3908	8,5262	0,1229	6,9609	4,7562
1° Quartil	7,9394	10,8560	2,7560	10,3115	8,2794
Mediana	9,1021	11,4425	4,9276	11,5188	9,5580
3° Quartil	12,1564	12,3441	7,9979	12,9201	11,1349
Máximo	16,2827	17,9164	14,0275	15,9492	13,5140

(b) Bent Cigar.

Estatística	BS	QN	AP	CC	APCC
Mínimo	9,4077	9,0786	3,1590	9,4543	5,6638
1° Quartil	10,7295	10,6969	6,5421	10,6681	8,3251
Mediana	11,0310	11,0275	9,1985	10,9775	9,4329
3° Quartil	11,3662	11,4033	10,8379	11,3160	10,8283
Máximo	12,5833	12,5037	12,5924	12,6895	12,6197

(c) Discus.

Estatística	BS	QN	AP	CC	APCC
Mínimo	0,0000	0,1051	0,0000	0,1597	0,4396
1° Quartil	3,5837	3,9411	3,5824	3,5981	3,7639
Mediana	3,6128	4,2890	3,6109	3,6291	4,1016
3° Quartil	3,6532	4,7596	3,7994	3,7604	4,2840
Máximo	4,8878	7,4385	5,4739	4,8323	6,4610

(d) Rosenbrock.

Estatística	BS	QN	AP	CC	APCC
Mínimo	3,0443	3,0473	3,0340	3,0443	3,0445
1° Quartil	3,0445	3,0654	3,0445	3,0445	3,0449
Mediana	3,0446	3,0706	3,0446	3,0446	3,0461
3° Quartil	3,0516	3,0751	3,0573	3,0500	3,0743
Máximo	3,1314	3,1291	3,1324	3,1290	3,1368

(e) Ackley.

Estatística	BS	QN	AP	CC	APCC
Mínimo	1,9808	1,9945	1,6102	1,7979	1,8367
1° Quartil	2,6470	2,6165	2,5501	2,6335	2,4231
Mediana	2,8306	2,7114	2,7603	2,7711	2,5766
3° Quartil	2,9318	2,8013	2,9218	2,9027	2,7528
Máximo	3,2099	3,1059	3,2224	3,2983	3,1002

(f) Weierstrass.

Estatística	BS	QN	AP	CC	APCC
Mínimo	0,0339	0,2963	0,0000	0,0974	0,1634
1° Quartil	0,1040	0,5072	0,0000	0,2522	0,4455
Mediana	0,1859	0,5979	0,0000	0,4082	0,5568
3° Quartil	0,3580	0,7394	0,0442	0,6140	0,7172
Máximo	1,3258	3,5731	0,6459	1,0221	1,4541

(g) Griewank.

Estatística	BS	QN	AP	CC	APCC
Mínimo	2,9792	3,4305	3,3624	3,3625	2,2195
1° Quartil	4,7349	4,4502	4,6983	4,7136	3,9069
Mediana	5,0544	4,7354	4,9922	5,0056	4,3627
3° Quartil	5,2799	5,0201	5,2213	5,2270	4,6597
Máximo	5,8848	5,7620	5,7633	5,9691	5,6175

(h) Rastrigin.

Estatística	BS	QN	AP	CC	APCC
Mínimo	5,8385	6,3033	6,3940	5,9351	6,1860
1° Quartil	7,2610	7,2141	7,2551	7,2668	7,1512
Mediana	7,4453	7,4464	7,4714	7,4753	7,3927
3° Quartil	7,6293	7,6114	7,6395	7,6380	7,5505
Máximo	7,9861	8,2156	7,9818	7,9937	7,9547

(i) Schwefel.

Estatística	BS	QN	AP	CC	APCC
Mínimo	0,1972	0,2821	0,1066	0,0426	0,2256
1° Quartil	0,6334	0,9257	0,6378	0,5099	0,8500
Mediana	0,9375	1,1387	0,9138	0,7837	1,1511
3° Quartil	1,2642	1,3450	1,2264	1,1320	1,4664
Máximo	2,4095	2,2242	2,3258	2,5569	2,7143

(j) Katsuura.

Estatística	BS	QN	AP	CC	APCC
Mínimo	0,1427	0,0557	0,0476	0,1589	0,0639
1° Quartil	0,3108	0,2682	0,2215	0,3239	0,2656
Mediana	0,4061	0,3698	0,2982	0,4182	0,3767
3° Quartil	0,5050	0,6783	0,4248	0,5204	0,5218
Máximo	3,3584	2,8351	3,0089	2,5066	2,6585

(k) HappyCat.

Estatística	BS	QN	AP	CC	APCC
Mínimo	0,1105	0,1320	0,1877	0,1098	0,1455
1° Quartil	0,2668	0,3386	0,3777	0,2669	0,3047
Mediana	0,3279	0,3936	0,3973	0,3540	0,3579
3° Quartil	0,6191	0,6139	0,4313	0,6176	0,5464
Máximo	1,6062	5,1274	0,9117	1,0503	4,9014

(l) HGBat.

Tabela B.4: Valores-p geral e das comparações pareadas entre os algoritmos pelo teste de Quade em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.

	BS	QN	AP	CC
QN	0,713104	–	–	–
AP	0,000000	0,000000	–	–
CC	0,000000	0,000001	0,000000	–
APCC	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000				

(a) Elliptic.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,000000	0,000000	–	–
CC	0,000000	0,029035	0,000000	–
APCC	0,087352	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000				

(b) Bent Cigar.

	BS	QN	AP	CC
QN	0,768788	–	–	–
AP	0,000000	0,000000	–	–
CC	0,804621	0,768788	0,000000	–
APCC	0,000000	0,000000	0,002769	0,000000
Valor-p do Teste: 0,000000				

(c) Discus.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,237702	0,000000	–	–
CC	0,004761	0,000000	0,094656	–
APCC	0,000000	0,000002	0,000000	0,000000
Valor-p do Teste: 0,000000				

(d) Rosenbrock.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,018509	0,000000	–	–
CC	0,028128	0,000000	0,000007	–
APCC	0,000000	0,000338	0,000000	0,000000
Valor-p do Teste: 0,000000				

(e) Ackley.

	BS	QN	AP	CC
QN	0,000043	–	–	–
AP	0,000552	0,527640	–	–
CC	0,145693	0,044990	0,145693	–
APCC	0,000000	0,000001	0,000000	0,000000
Valor-p do Teste: 0,000000				

(f) Weierstrass.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,000000	0,000000	–	–
CC	0,000006	0,000000	0,000000	–
APCC	0,000000	0,000049	0,000000	0,000000
Valor-p do Teste: 0,000000				

(g) Griewank.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,389696	0,000003	–	–
CC	0,590855	0,000000	0,639216	–
APCC	0,000000	0,000000	0,000000	0,000000
Valor-p do Teste: 0,000000				

(h) Rastrigin.

	BS	QN	AP	CC
QN	1,000000	–	–	–
AP	1,000000	1,000000	–	–
CC	1,000000	1,000000	1,000000	–
APCC	0,054653	0,223515	0,057617	0,077037
Valor-p do Teste: 0,026840				

(i) Schwefel.

	BS	QN	AP	CC
QN	0,000006	–	–	–
AP	1,000000	0,000001	–	–
CC	0,025644	0,000000	0,066237	–
APCC	0,000024	1,000000	0,000004	0,000000
Valor-p do Teste: 0,000000				

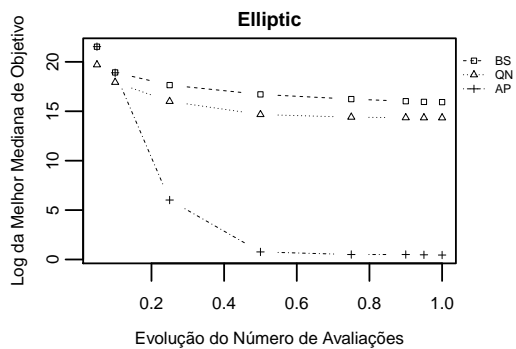
(j) Katsuura.

	BS	QN	AP	CC
QN	0,010313	–	–	–
AP	0,008076	0,000000	–	–
CC	0,699096	0,048755	0,001164	–
APCC	0,699096	0,000440	0,081331	0,409645
Valor-p do Teste: 0,000000				

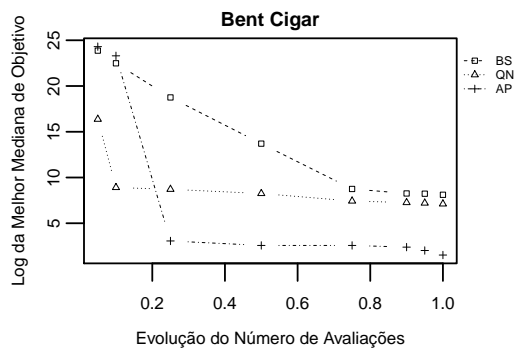
(k) HappyCat.

	BS	QN	AP	CC
QN	0,000000	–	–	–
AP	0,548458	0,000102	–	–
CC	1,000000	0,000001	1,000000	–
APCC	1,000000	0,000016	1,000000	1,000000
Valor-p do Teste: 0,000000				

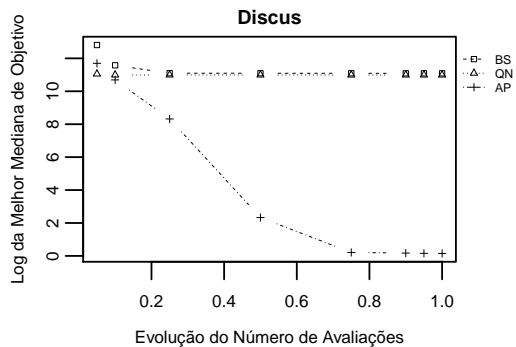
(l) HGBat.



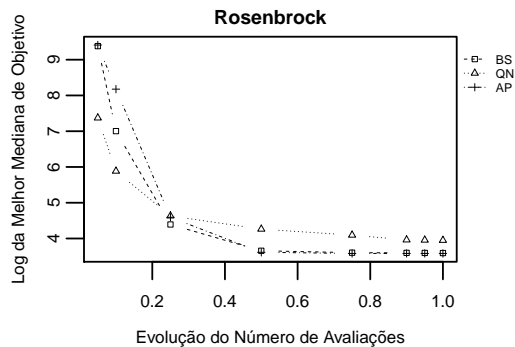
(a) Elliptic.



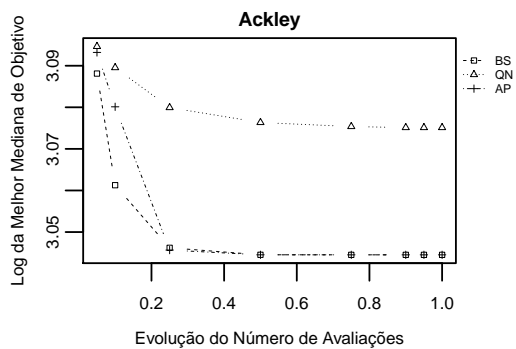
(b) Bent Cigar.



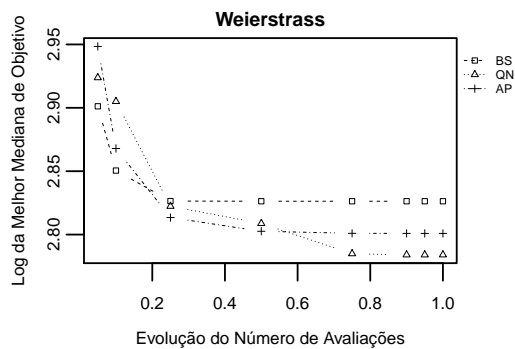
(c) Discus.



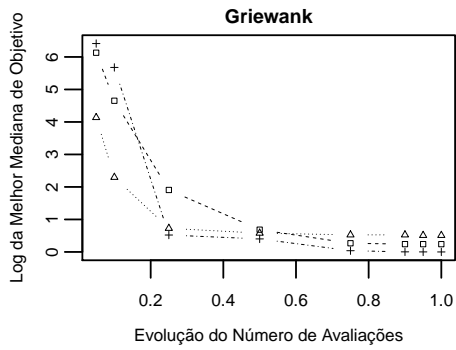
(d) Rosenbrock.



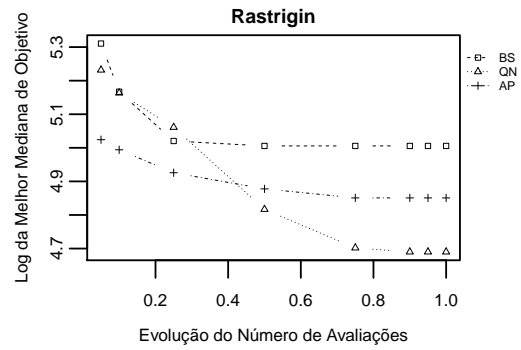
(e) Ackley.



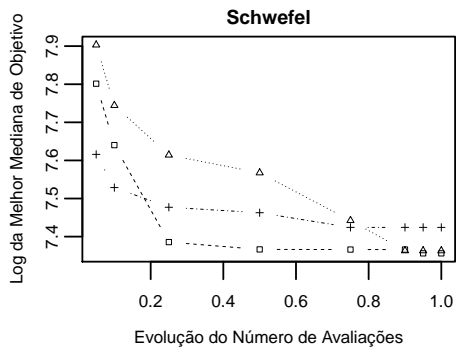
(f) Weierstrass.



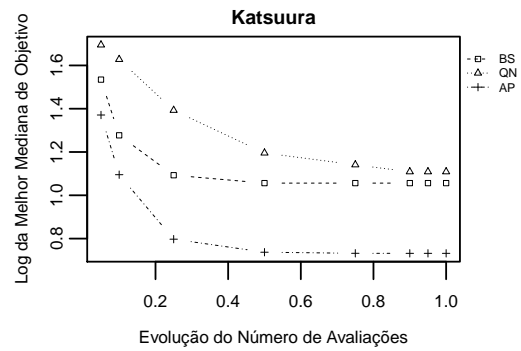
(g) Griewank.



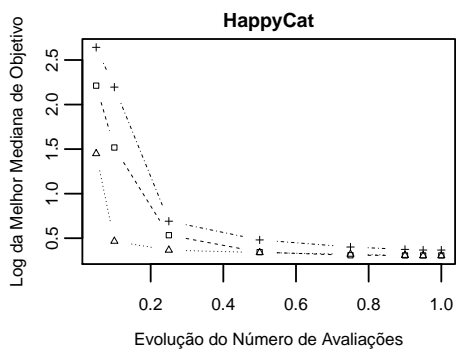
(h) Rastrigin.



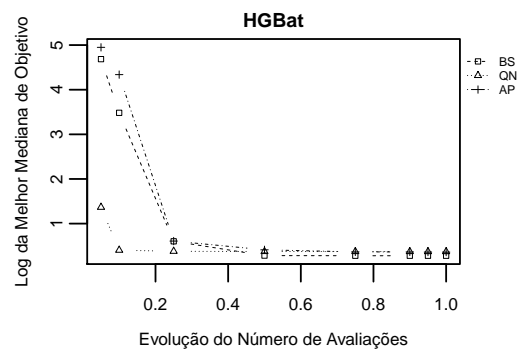
(i) Schwefel.



(j) Katsuura.

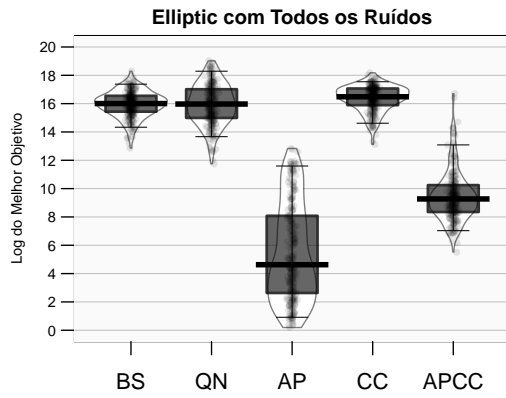


(k) HappyCat.

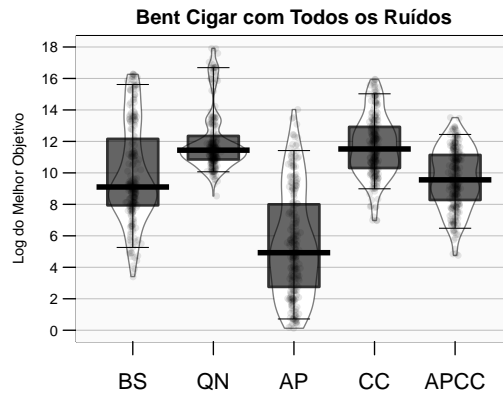


(l) HGBat.

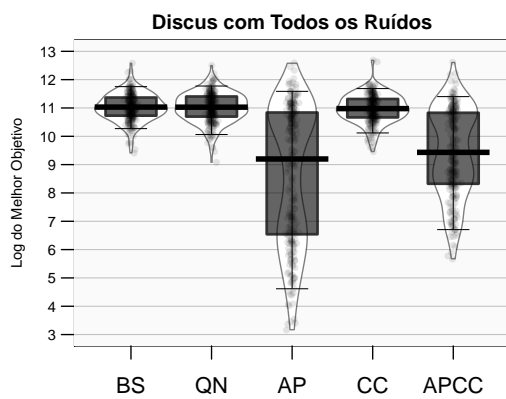
Figura B.2: Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo obtido pelos algoritmos pela progressão do número de avaliações, em relação à cada função-objetivo no experimento sem ruído.



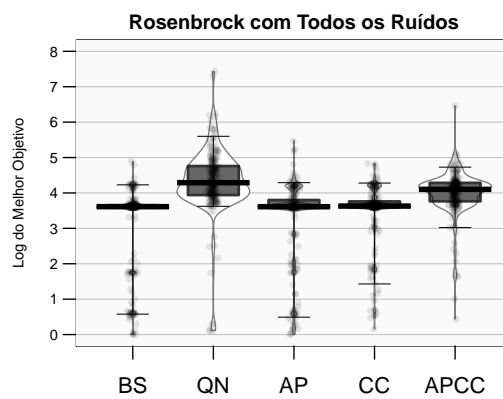
(a) Elliptic.



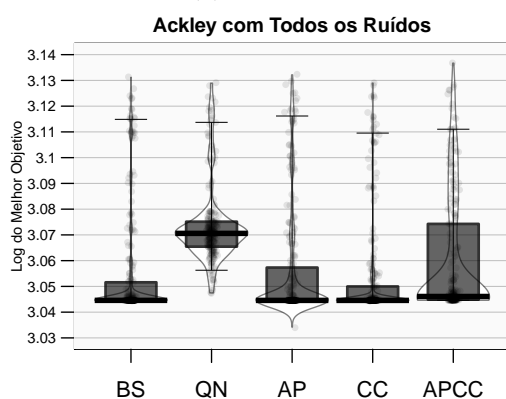
(b) Bent Cigar.



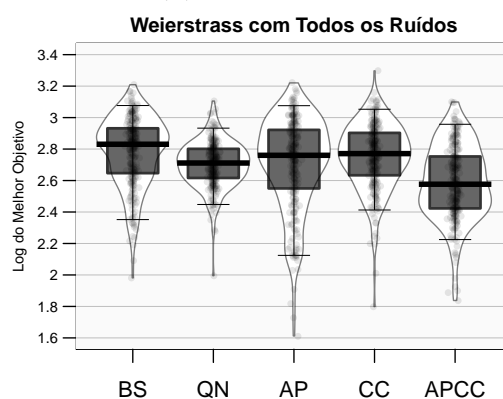
(c) Discus.



(d) Rosenbrock.



(e) Ackley.



(f) Weierstrass.

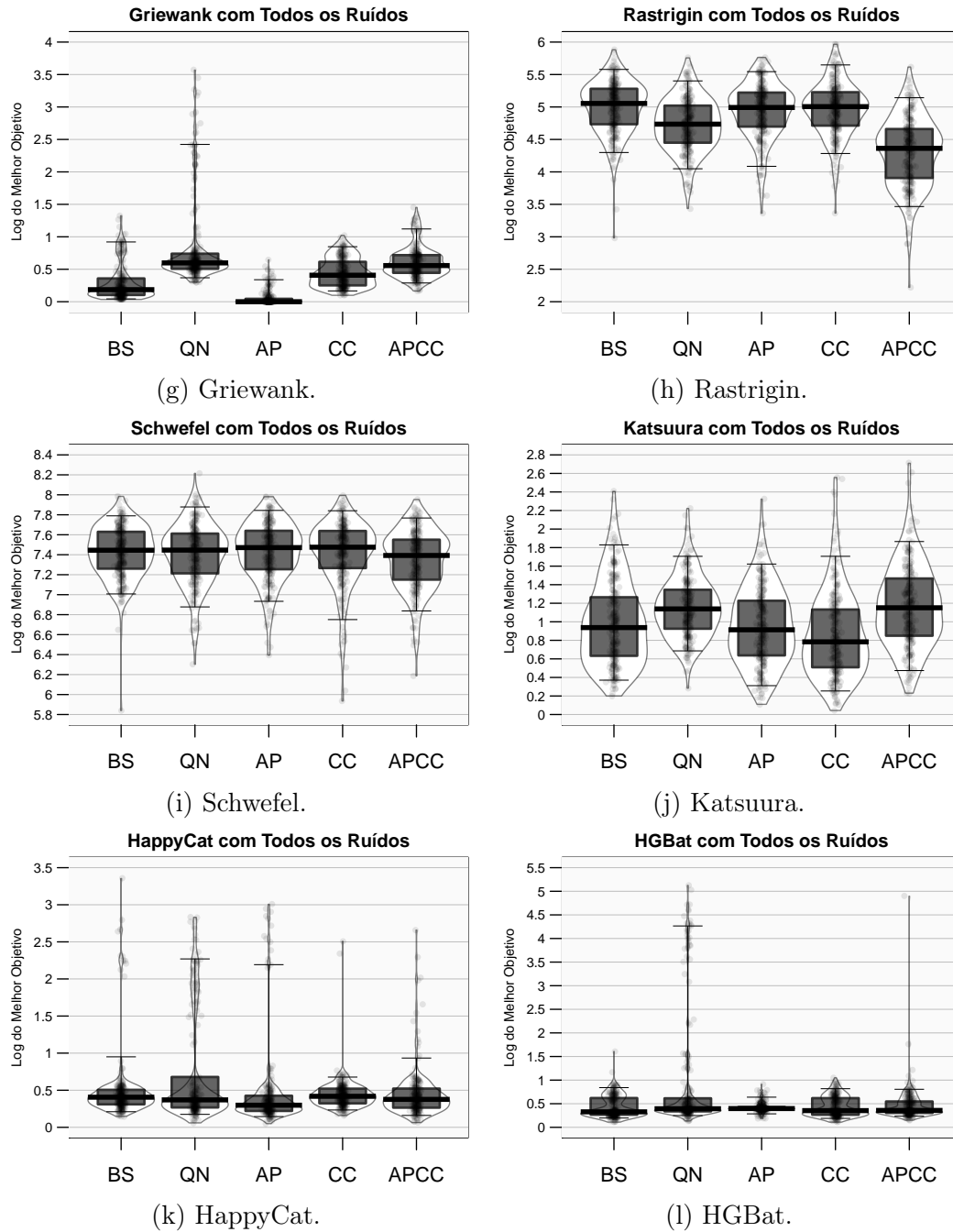
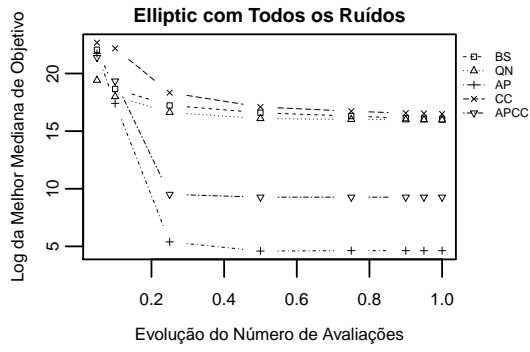
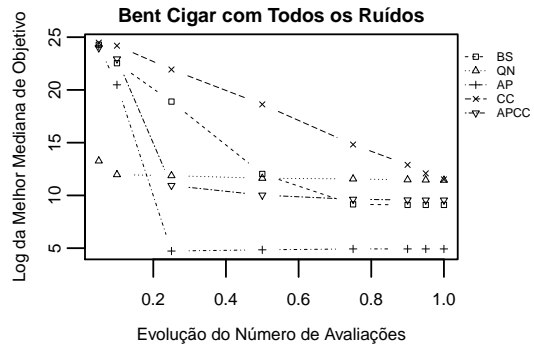


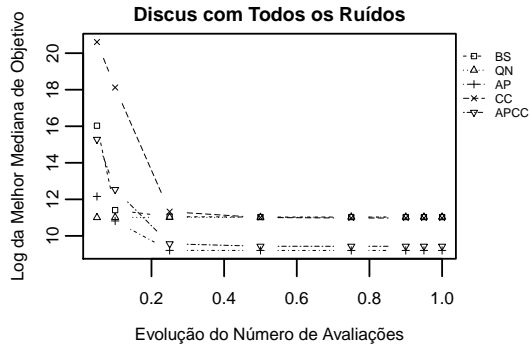
Figura B.3: Beanplot para o $\log(f(x) + 1)$ dos melhores valores de objetivo reais retornados pelos algoritmos em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.



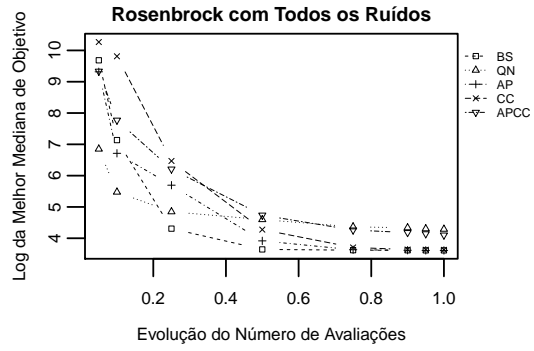
(a) Elliptic.



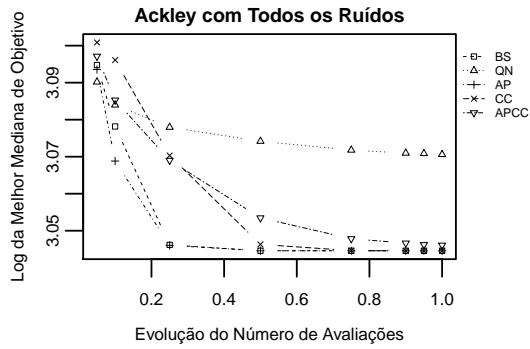
(b) Bent Cigar.



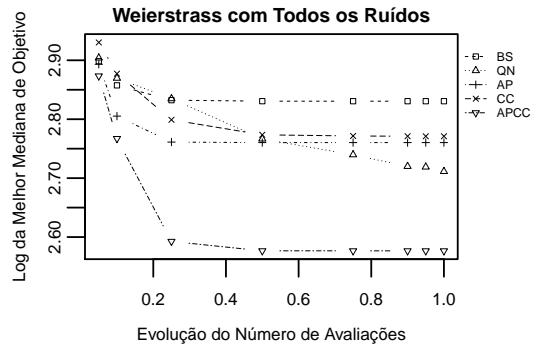
(c) Discus.



(d) Rosenbrock.



(e) Ackley.



(f) Weierstrass.

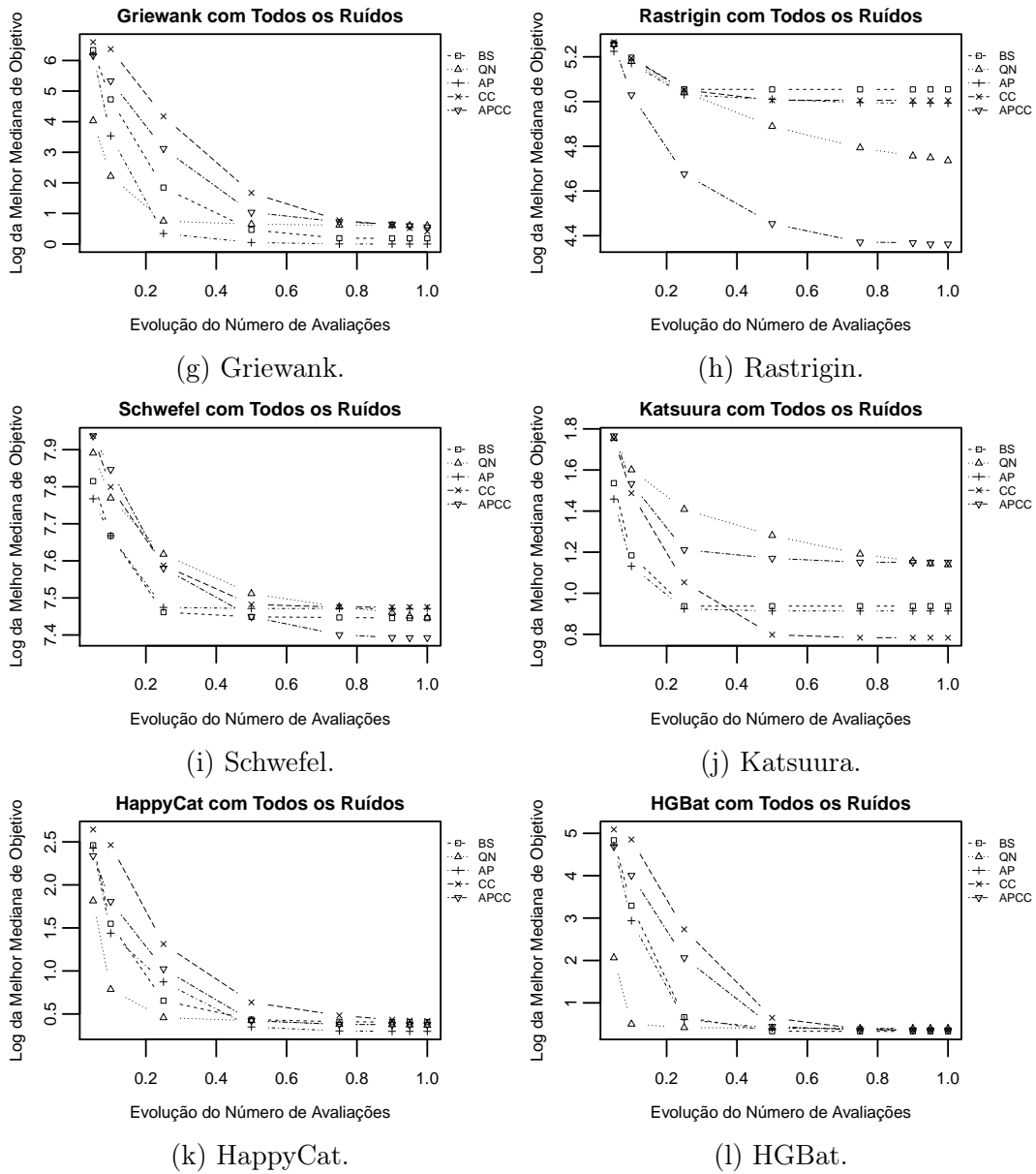


Figura B.4: Evolução da mediana do $\log(f(x) + 1)$ para o melhor valor de objetivo real obtido pelos algoritmos pela progressão do número de avaliações, em relação à cada função-objetivo ruidosa e todos os níveis de desvio padrão.

Tabela B.5: Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-BS nas 30 execuções.

Taxa 1	Taxa 2	Taxa 3	Taxa 4	Taxa 5	Taxa 6	Taxa 7	Taxa 8	Taxa 9	Taxa 10
0,8964	0,8987	0,3523	0,1825	0,7236	0,7567	0,8554	0,2822	0,5604	0,9054
0,9139	0,6088	0,3159	0,2026	0,6510	0,5982	0,9415	0,7166	0,8555	0,9497
0,9411	0,4585	0,6648	0,5577	0,3083	0,8840	0,4471	0,9451	0,9145	0,7884
0,7698	0,6355	0,4175	0,9381	0,9030	0,5202	0,7971	0,1643	0,3230	0,9463
0,9500	0,7616	0,7623	0,5387	0,2014	0,1763	0,9465	0,9492	0,7764	0,8617
0,9286	0,9052	0,1976	0,3530	0,8722	0,8571	0,9363	0,7201	0,1339	0,3216
0,9427	0,0500	0,9461	0,4357	0,0909	0,9401	0,5606	0,8952	0,9450	0,9435
0,9430	0,2567	0,9361	0,1779	0,0990	0,5235	0,1647	0,4521	0,1368	0,9500
0,9500	0,4489	0,3340	0,8413	0,9500	0,8094	0,7158	0,8494	0,4102	0,0523
0,9293	0,1842	0,8437	0,5657	0,8275	0,3838	0,9423	0,6444	0,9134	0,2389
0,8969	0,9359	0,5347	0,0914	0,4749	0,8316	0,6845	0,9349	0,9302	0,9256
0,1128	0,9364	0,9165	0,6517	0,4433	0,7118	0,8844	0,0659	0,9442	0,8957
0,9340	0,9137	0,8752	0,9500	0,9071	0,4135	0,0647	0,0533	0,0519	0,0503
0,9479	0,8637	0,6178	0,3276	0,7597	0,0557	0,9500	0,9326	0,7622	0,8528
0,7883	0,9205	0,1985	0,0993	0,4513	0,1025	0,1156	0,5052	0,7093	0,9340
0,8146	0,8443	0,5591	0,0769	0,5448	0,9200	0,4165	0,1225	0,0521	0,9089
0,1117	0,6151	0,1329	0,3634	0,2661	0,3718	0,9406	0,0515	0,7262	0,9362
0,7229	0,8579	0,9498	0,0616	0,8474	0,2083	0,9138	0,7107	0,2163	0,9494
0,8516	0,1034	0,6215	0,6663	0,6042	0,2032	0,9298	0,3940	0,9427	0,9122
0,9421	0,9208	0,7726	0,8715	0,4821	0,9079	0,9369	0,3783	0,5928	0,8086
0,8983	0,6203	0,9423	0,4448	0,4513	0,9008	0,8976	0,2258	0,7339	0,0556
0,9486	0,4222	0,9303	0,3790	0,8863	0,2314	0,9496	0,8517	0,8558	0,9471
0,9341	0,8846	0,7774	0,8951	0,5722	0,5891	0,1429	0,9351	0,8524	0,1237
0,8473	0,0503	0,7652	0,1713	0,0502	0,2562	0,0809	0,2769	0,9242	0,8569
0,9443	0,9497	0,2091	0,2845	0,5319	0,3387	0,0735	0,4268	0,5783	0,1658
0,9500	0,9329	0,5446	0,9121	0,8307	0,3192	0,7129	0,8281	0,0600	0,2716
0,9056	0,0831	0,2765	0,1135	0,0809	0,0599	0,5907	0,1496	0,6162	0,9382
0,9126	0,9500	0,5886	0,9197	0,7210	0,9208	0,1188	0,2261	0,4027	0,0536
0,9488	0,7578	0,0774	0,6525	0,9008	0,6971	0,5501	0,8615	0,6947	0,9500
0,9462	0,8466	0,1101	0,7996	0,0511	0,0574	0,7483	0,9376	0,9210	0,3829

Tabela B.6: Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-QN nas 30 execuções.

Taxa 1	Taxa 2	Taxa 3	Taxa 4	Taxa 5	Taxa 6	Taxa 7	Taxa 8	Taxa 9	Taxa 10
0,8960	0,8984	0,8543	0,0849	0,9499	0,5396	0,7132	0,8571	1,2733	0,5898
0,7912	1,2628	0,8364	0,6510	0,9481	0,3860	0,9431	0,7109	0,0509	0,6958
0,8950	0,8352	0,8652	0,3444	0,0671	0,9467	0,9499	0,0634	1,2641	0,3337
1,0721	0,8697	0,6108	0,0809	0,5360	0,9410	0,0578	0,7870	0,1736	0,8915
1,2299	0,2339	0,4875	0,5328	0,4118	0,6158	0,7596	0,1579	0,1406	0,8941
1,0720	0,0540	0,0520	0,8903	0,5715	0,2487	0,7361	0,9433	0,9162	0,8805
1,1777	0,6207	0,1987	0,8266	0,0537	0,4609	0,1352	0,2859	0,7983	0,9462
1,2346	0,8389	0,0717	0,1267	0,3723	0,1349	0,9125	0,8659	0,8729	0,9165
0,9462	0,8369	0,7113	0,0876	0,6093	0,7759	0,5406	0,0612	0,7186	0,9024
0,6362	0,8283	0,9307	0,0509	0,1744	0,1224	0,8627	0,9422	0,2797	1,1530
1,2225	0,5938	0,5646	0,7787	0,1363	0,4590	0,9313	0,8062	0,6590	0,9222
0,9021	0,2969	0,4589	0,5706	0,1358	0,3607	0,2502	0,3125	0,4757	0,9361
0,8979	0,9381	0,7502	0,4563	0,5250	0,0515	0,8854	0,5407	0,2985	0,9483
1,1431	0,5262	0,7285	0,8564	0,3963	0,0984	0,7686	0,7717	0,6896	0,8411
0,9475	0,0548	0,8475	0,3576	0,1284	0,4771	0,8457	0,1062	0,7620	0,9476
1,2089	0,5733	0,4238	0,2988	0,0713	0,6330	0,4623	0,5570	0,1492	0,9225
1,0632	0,8759	0,8108	0,8123	0,7781	0,8876	0,3365	0,9357	0,7885	0,7522
1,2670	0,0777	0,9436	0,7019	0,2373	0,7110	0,4923	0,1402	0,2086	0,9398
0,9774	0,9279	0,3803	0,6595	0,7084	0,1790	0,7485	0,0522	0,1322	0,8437
1,2621	0,2412	0,6559	0,1655	0,9227	0,7566	0,3374	0,0518	0,0545	0,9235
1,1092	0,0957	0,0847	0,7202	0,1816	0,8384	0,6220	0,8904	0,5602	0,8952
0,4547	0,5872	0,1148	0,1976	0,9111	0,7527	0,7758	0,6789	0,5156	0,8907
1,2578	0,9228	0,4101	0,2516	0,3874	0,8833	0,8904	0,4905	0,1567	0,9347
1,2539	0,0616	0,6310	0,3939	0,5986	0,8754	0,5905	0,0558	0,5750	0,9349
0,8874	0,5474	0,8011	0,2547	0,5276	0,8896	0,2890	0,3371	0,2030	0,9349
0,9983	0,0513	0,6537	0,4657	0,7684	0,8986	0,8628	0,4386	0,4698	0,9465
1,1898	0,3817	0,5732	0,7028	0,3451	0,1944	0,8621	0,5933	0,0880	0,4763
1,2529	0,1700	0,7624	0,3289	0,0515	0,5952	0,4450	0,8559	0,8438	0,9407
0,8933	0,7734	0,4307	1,2358	0,8874	0,7934	0,8593	0,0560	0,9221	0,8200
0,9267	0,5285	0,7949	0,1614	1,2238	0,9231	0,0618	0,9013	0,0864	0,6634

Tabela B.7: Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-AP nas 30 execuções.

Taxa 1	Taxa 2	Taxa 3	Taxa 4	Taxa 5	Taxa 6	Taxa 7	Taxa 8	Taxa 9	Taxa 10
0,7508	0,4810	0,5215	0,9450	0,9314	0,4350	0,6501	0,9273	0,7964	0,9310
0,9494	0,9494	0,4730	0,8987	0,1503	0,0507	0,1037	0,6544	0,9295	0,1142
0,9492	0,4526	0,4152	0,9354	0,5754	0,2164	0,9445	0,7573	0,4623	0,8220
0,9114	0,4309	0,4106	0,8479	0,9432	0,7473	0,3342	0,8876	0,8780	0,1435
0,9478	0,2286	0,8226	0,0527	0,8824	0,2469	0,6776	0,0508	0,9319	0,7665
0,9492	0,9488	0,1315	0,9092	0,4950	0,8880	0,9471	0,8942	0,8157	0,0506
0,9331	0,0586	0,6503	0,1091	0,9385	0,8462	0,4363	0,3913	0,8890	0,9489
0,9500	0,6987	0,9091	0,4316	0,4471	0,7791	0,8968	0,3161	0,8574	0,9500
0,9336	0,9427	0,7636	0,2989	0,9424	0,9227	0,7721	0,5761	0,7644	0,0501
0,9379	0,4851	0,1310	0,9462	0,0904	0,8958	0,0546	0,0739	0,8665	0,9500
0,9175	0,6780	0,9059	0,3851	0,9378	0,7140	0,9463	0,6348	0,7945	0,7261
0,9436	0,8775	0,8536	0,5543	0,6479	0,8571	0,7907	0,8155	0,7904	0,7866
0,9495	0,4546	0,5948	0,9364	0,7623	0,4178	0,6189	0,9186	0,5172	0,2213
0,9490	0,6933	0,8533	0,2413	0,1497	0,3967	0,5086	0,8095	0,0527	0,9500
0,9499	0,8945	0,7167	0,8945	0,9479	0,0666	0,7418	0,1564	0,7199	0,4863
0,9314	0,9497	0,4140	0,8141	0,9202	0,5048	0,7027	0,7980	0,9422	0,3234
0,9498	0,2148	0,7596	0,7689	0,2813	0,9328	0,8314	0,8792	0,7071	0,8491
0,9323	0,0941	0,0726	0,7767	0,7294	0,3494	0,5769	0,3427	0,5266	0,9260
0,8613	0,7424	0,3769	0,6818	0,3090	0,6709	0,5224	0,9496	0,1095	0,6743
0,9499	0,9413	0,2613	0,8986	0,5007	0,6495	0,6508	0,6684	0,9087	0,6751
0,9019	0,2008	0,6124	0,5799	0,9065	0,3500	0,1312	0,0674	0,8877	0,9480
0,9462	0,9498	0,1477	0,8450	0,9434	0,3448	0,0544	0,9444	0,2119	0,0551
0,8775	0,5581	0,9186	0,6636	0,7689	0,2629	0,5267	0,9446	0,1726	0,9354
0,9498	0,8179	0,1896	0,1419	0,1750	0,0543	0,4733	0,5439	0,4950	0,2149
0,9499	0,6994	0,9500	0,6218	0,5496	0,4301	0,4783	0,5038	0,3578	0,9353
0,8950	0,9155	0,3788	0,2876	0,5645	0,8991	0,3572	0,8838	0,4546	0,9099
0,9117	0,8734	0,9482	0,6665	0,8529	0,9349	0,6996	0,6078	0,8606	0,0532
0,9254	0,2298	0,8288	0,9421	0,7509	0,4197	0,7517	0,5177	0,1126	0,7287
0,9209	0,9265	0,5891	0,8634	0,9328	0,8688	0,2484	0,8945	0,6359	0,0525
0,9363	0,3771	0,8638	0,9201	0,4859	0,8074	0,0889	0,5170	0,8677	0,9481

Tabela B.8: Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-CC nas 30 execuções.

Taxa 1	Taxa 2	Taxa 3	Taxa 4	Taxa 5	Taxa 6	Taxa 7	Taxa 8	Taxa 9	Taxa 10
0,9431	0,7930	0,7245	0,4829	0,8564	0,8090	0,9470	0,9355	0,1854	0,5385
0,9291	0,3787	0,9359	0,6921	0,9453	0,2615	0,6883	0,7339	0,8636	0,9325
0,8420	0,9379	0,4622	0,0527	0,8567	0,0520	0,8806	0,8039	0,9189	0,9489
0,9500	0,9449	0,0730	0,5386	0,5003	0,6848	0,3242	0,4273	0,8247	0,9471
0,8095	0,9498	0,8162	0,9065	0,4686	0,6098	0,8445	0,7328	0,8660	0,4356
0,9201	0,4393	0,5999	0,0500	0,2621	0,5736	0,1197	0,8434	0,3035	0,9487
0,9486	0,0527	0,7385	0,6168	0,9477	0,8807	0,6593	0,6799	0,6642	0,9342
0,9343	0,1865	0,3611	0,3497	0,8621	0,9445	0,9263	0,8188	0,8012	0,9030
0,7691	0,5528	0,7182	0,0777	0,1521	0,9432	0,4266	0,1792	0,1294	0,9499
0,9500	0,8352	0,4806	0,1550	0,8325	0,4583	0,6749	0,9310	0,9258	0,9486
0,9493	0,2822	0,0501	0,7514	0,9260	0,7583	0,0529	0,3115	0,5736	0,9500
0,9477	0,9210	0,7082	0,9423	0,9001	0,9483	0,9103	0,5430	0,8456	0,6102
0,9323	0,2745	0,0636	0,7059	0,4936	0,7996	0,3994	0,7604	0,7042	0,9400
0,9188	0,4118	0,4792	0,6611	0,7955	0,9486	0,1372	0,9066	0,2268	0,9496
0,9226	0,4171	0,1668	0,9497	0,9012	0,8014	0,4981	0,6367	0,5904	0,9499
0,9484	0,9444	0,9474	0,4374	0,6141	0,9491	0,0892	0,8209	0,6724	0,9424
0,9495	0,7673	0,3518	0,8651	0,3122	0,5796	0,0521	0,8993	0,3998	0,9500
0,9360	0,4829	0,9500	0,7043	0,4047	0,9499	0,6800	0,8495	0,7123	0,7855
0,8028	0,7950	0,2546	0,7124	0,4627	0,3202	0,1841	0,7616	0,0572	0,8828
0,9488	0,0894	0,9410	0,8003	0,9361	0,8285	0,8763	0,0526	0,8538	0,8492
0,9494	0,3274	0,2532	0,4453	0,6717	0,5302	0,2670	0,9442	0,8916	0,9500
0,9499	0,6805	0,9252	0,5342	0,7685	0,9457	0,6764	0,8279	0,6750	0,8504
0,9500	0,2513	0,8903	0,3771	0,0523	0,5003	0,9298	0,8998	0,9491	0,9375
0,9467	0,6338	0,3119	0,3544	0,8998	0,8873	0,4897	0,9440	0,9286	0,2848
0,9297	0,8277	0,5373	0,7203	0,9294	0,9175	0,3927	0,9359	0,3544	0,9289
0,9498	0,9120	0,5945	0,9216	0,3698	0,7165	0,9327	0,6663	0,2048	0,8493
0,8307	0,9200	0,5483	0,9467	0,5805	0,9060	0,9244	0,9353	0,8581	0,8204
0,8885	0,7866	0,1682	0,8374	0,9499	0,8769	0,9497	0,9398	0,5028	0,8325
0,9395	0,7966	0,6607	0,1275	0,0962	0,8738	0,2810	0,8003	0,1473	0,9498
0,9500	0,9061	0,2903	0,2312	0,4816	0,8924	0,9226	0,1828	0,6573	0,9385

Tabela B.9: Melhores de taxas de vacinação de suscetíveis retornadas pelo algoritmo ES-APCC nas 30 execuções.

Taxa 1	Taxa 2	Taxa 3	Taxa 4	Taxa 5	Taxa 6	Taxa 7	Taxa 8	Taxa 9	Taxa 10
0,9423	0,3157	0,1871	0,6264	0,7577	0,0704	0,1424	0,5481	0,7046	0,9308
0,8852	0,5153	0,8892	0,2077	0,8920	0,1206	0,4321	0,8527	0,1715	0,9315
0,9357	0,8379	0,7910	0,0702	0,9086	0,1027	0,2622	0,2417	0,3960	0,9023
0,9442	0,8841	0,9072	0,4128	0,7539	0,9275	0,7407	0,6345	0,7306	0,8688
0,9349	0,8129	0,9034	0,6156	0,5742	0,9317	0,3804	0,1440	0,5023	0,9142
0,9498	0,5517	0,7469	0,8818	0,0620	0,5822	0,1741	0,7890	0,9500	0,8880
0,9296	0,2193	0,1065	0,8614	0,7078	0,6945	0,8907	0,9060	0,6986	0,9497
0,9358	0,9383	0,0584	0,1757	0,6833	0,9460	0,5098	0,0990	0,8959	0,9500
0,9487	0,4951	0,5616	0,3256	0,9279	0,7831	0,1328	0,7694	0,7392	0,9424
0,7540	0,5390	0,3352	0,9225	0,0714	0,0519	0,0502	0,7680	0,8982	0,9500
0,9355	0,0525	0,9417	0,4900	0,8047	0,3027	0,8738	0,2130	0,0841	0,9444
0,9420	0,2991	0,8836	0,3568	0,4256	0,8166	0,6585	0,8618	0,5723	0,9284
0,9494	0,6104	0,7069	0,8406	0,8518	0,9248	0,9158	0,0672	0,8152	0,7860
0,9230	0,8290	0,8744	0,1265	0,8893	0,0500	0,0913	0,5012	0,7519	0,9444
0,9499	0,0612	0,0509	0,7220	0,2604	0,7715	0,5783	0,5636	0,2191	0,9499
0,9407	0,4356	0,5917	0,7530	0,7816	0,8483	0,6551	0,9415	0,2370	0,9454
0,9407	0,8432	0,3274	0,2364	0,2101	0,6616	0,2555	0,9471	0,7649	0,9244
0,9373	0,8385	0,8695	0,8723	0,8656	0,0556	0,2410	0,6064	0,0871	0,9439
0,4364	0,3745	0,0601	0,7980	0,4740	0,6171	0,8560	0,2122	0,7528	0,9471
0,9456	0,7926	0,2759	0,6686	0,6607	0,5819	0,6952	0,9390	0,9458	0,9494
0,9498	0,5853	0,0897	0,1354	0,3492	0,5234	0,5769	0,4935	0,2720	0,9457
0,8950	0,0841	0,7375	0,4853	0,5873	0,9461	0,0826	0,6592	0,8156	0,9431
0,6470	0,5787	0,7375	0,1089	0,3904	0,1129	0,5828	0,3461	0,2835	0,8988
0,9493	0,9190	0,2239	0,4804	0,6561	0,7847	0,6494	0,3759	0,0917	0,8917
0,8580	0,6517	0,9317	0,9053	0,2048	0,9364	0,1624	0,8232	0,7548	0,9258
0,9289	0,8845	0,0564	0,6743	0,8779	0,3122	0,6937	0,0711	0,8786	0,9487
0,9409	0,8105	0,8774	0,6999	0,7763	0,6717	0,4267	0,8873	0,8818	0,9389
0,9351	0,7578	0,0642	0,6519	0,8885	0,9498	0,5251	0,9033	0,2878	0,9474
0,9294	0,0809	0,8652	0,8976	0,8058	0,8778	0,6477	0,9354	0,4315	0,9448
0,8744	0,9157	0,3299	0,8469	0,6502	0,1725	0,7907	0,0566	0,6288	0,9499

Apêndice C

Multiobjective synthesis of robust vaccination policies



Multiobjective synthesis of robust vaccination policies



André R. da Cruz^{a,b}, Rodrigo T.N. Cardoso^c, Ricardo H.C. Takahashi^{d,*}

^a Dep. de Computação e Construção Civil – Centro Federal de Educação Tecnológica de Minas Gerais, R. Dezenove de Novembro, 121, Centro, Timóteo, MG, Brazil

^b Programa de Pós-Graduação em Engenharia Elétrica – Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG, Brazil

^c Dep. de Física e Matemática – Centro Federal de Educação Tecnológica de Minas Gerais, Av. Amazonas, 7675, Nova Gameleira, Belo Horizonte, MG, Brazil

^d Dep. de Matemática – Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627, Pampulha, Belo Horizonte, MG, Brazil

ARTICLE INFO

Article history:

Received 5 February 2016

Received in revised form

17 September 2016

Accepted 7 November 2016

Available online 14 November 2016

Keywords:

Multiobjective optimization

Vaccination planning

Robust synthesis

Epidemics control

Genetic algorithms

ABSTRACT

This paper deals with the optimal planning of vaccination campaigns, using an evolutionary multiobjective optimization algorithm and a stochastic simulation of the epidemics dynamics in order to determine robust vaccination policies. A biobjective model is formulated, considering the minimization of control costs and number of infected individuals. The decision variables include number of campaigns, percentage of vaccination and time interval between each campaign. A SIR (Susceptible-Infected-Recovered) model and an IBM (Individual-Based Model) are employed for representing the epidemics. A two-stage optimization process is proposed: a set of nondominated steady-state regimes is obtained and one of them is selected to be concatenated to the transient regime vaccination policies. An evolutionary multiobjective optimization algorithm is proposed, with a local search procedure based on quadratic approximation supported by a hash table information storage. The resulting nondominated solutions are simulated in the IBM, in order to detect and discard the non-robust solutions. Final results show that optimal robust vaccination campaigns with different trade-offs can be designed, allowing policymakers to choose the best strategy according to the monetary cost and the expected efficacy.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

An epidemic is defined as the occurrence of cases of a disease in excess of what is normally expected in a population. An endemic is a relatively stable pattern of occurrence of a transmissible disease in a population group with relatively high prevalence and incidence. A big challenge in public health is related to the planning of vaccination campaigns which aim to eradicate a disease, or to control its spread avoiding large epidemic peaks or significant endemic states, with the minimum possible cost [24]. This paper presents a multiobjective framework, considering a compromise between cost and effectiveness, to determine *nondominated* and *robust* policies to control a disease through impulsive vaccination. A nondominated solution is not worse than any other possible one in both cost and effectiveness, and a robust solution does not degrade significantly its performance when different realizations of the underlying stochastic processes are considered.

Many works have applied control theory to develop optimal strategies to control infectious diseases [25,28,26,27]. Most of them used pulse vaccination, defined as an impulsive control composed by the repeated application of a fixed vaccination ratio in discrete instants [2,20,12]. Stochastic programming frameworks and mono-objective optimization techniques have also been used [31–34]. Different methodologies to the optimization of vaccination campaigns can be found in the literature [29–31,26].

From the modelling perspective, some articles consider generic epidemic models (such as SIR) [28,25], while other ones use specific models of some specific epidemics such as influenza [29,31,32]. In [32], genetic algorithms were applied with specific operators and stochastic epidemic simulations to find good vaccine distributions to minimize the number of illnesses or deaths in the population, considering an influenza pandemic with age-specific illness attack, given limited quantities of vaccine. More recently, [26] proposed a mono-objective strategy using a Differential Evolution algorithm and a SEIR model with age-structure in order to find the optimal vaccine distribution that minimizes the total number of infected. The work [27] applied a standard multiobjective genetic algorithm to control dengue epidemic using insecticide and sterile males,

* Corresponding author.

E-mail address: taka@mat.ufmg.br (R.H.C. Takahashi).

minimizing both social and economic costs via alternated step-size control.

The trade-off between the effectiveness of policies to control the epidemics and the cost of its implementation has been already recognized in some other works, which already employed multi-objective optimization approaches for the synthesis of vaccination policies. This work, developed within this framework, also presents the following specific methodological contributions:

- (i) Every epidemics process will present two rather distinct dynamic regimes: a transient phase and a steady-state phase. Whilst the transient phase will require a time-varying control policy in order to perform an optimal control, the steady-state phase will be tackled better by a steady-state control, since its synthesis will become more precise under a smaller number of degrees of freedom. In this work, the control policy is built as a concatenation of a transient control sequence followed by a time-invariant sequence. This allows an enhanced efficiency of the control policy, as illustrated in Section 6. The usual methodology of promoting a single vaccination policy along all the duration of the epidemic process leads to policies that can be far from the optimality [24,23,12]. Compared to the previous works, the approach presented here adds new degrees of freedom to the problem by allowing different sizes of pulse control action in different instants and also allowing the application of pulses at arbitrary time instants in the two phases. Given a time horizon, a control policy solution is represented here by the number of vaccination campaigns, the time instants when each one should be performed as well as the number of individuals to be immunized in each campaign. The search space used in this work includes the intervals found in reference [20], which theoretically guarantee that the infected population tends to zero over time.
- (ii) The synthesis of control policy is performed using a differential equation based epidemics model, which represents the average behavior of the epidemics on the population. The simulation of this model is computationally affordable, which makes this kind of model suitable for being used inside an optimization algorithm in which the model is run several times. Although this work considers Susceptible-Infected-Recovered endemic model (SIR) [15,2,12], it should be noticed that the general methodology proposed here can be employed using other differential equation models.
- (iii) However, as the actual behavior of the epidemics on a finite-size population will be endowed with a stochastic variation, a simulation that takes such effects into account becomes necessary in order to perform a more realistic evaluation of the outcome of the application of the vaccination policy. An Individual-Based Model (IBM) [11] which corresponds in average to the differential equation model is employed here in order to evaluate the set of policies that emerge from the multi-objective optimization, in order to assess the sensitivity of each policy to the stochasticity. This model is also employed in order to estimate some stochastic performance measures of the vaccination policy, such as the probability of disease eradication. Such an analysis was not performed in any former work.

In addition to the new methodology, some computational improvements have been proposed too. The hybrid multiobjective strategy was improved with the inclusion of a hash table to avoid re-evaluating of repeated solutions and some modifications in genetic operators. Finally, an enhanced IBM version is applied to find a robust nondominated set of policies. The probabilities of disease control and extinction for each policy are computed now. The new framework to epidemics control proposed here can be seen as an evolution of some previous works by the authors. In

[6], a canonical NSGA-II was used to solve a multiobjective problem whose solutions were transient-phase vaccination campaigns. The objectives represented the cost and the efficacy of the control. Also, a rudimentary IBM version was applied to simulate the random behavior of the final solutions, re-evaluating the optimal policies in order to consider just their median values in a classical nondominated sorting procedure. In [8], the performance of an NSGA-II algorithm with quadratic local search was compared with the performance of its canonical version in order to solve the epidemic control problem, still considering only the transient-phase vaccination campaigns.

This paper is organized as follows: Section 2 describes the epidemiological models; Section 3 presents the multiobjective optimization models developed in this work; Section 4 shows the proposed optimization engine; Section 5 presents the experiment planning; Section 6 shows the results of the case study; before finalize this paper, Section 7 indicates that the proposed optimization engine is superior than the canonical version in many disease configurations; and Section 8 concludes the paper.

2. Epidemiological models

The next two subsections explain the SIR model and its stochastic version IBM.

2.1. SIR model

The SIR model describes the dynamics of the susceptible, infected and recovered individuals in a population during the evolution of the disease, in an average way. It can be used to describe contagious viruses that can be transmitted among individuals, like measles and rubella. The virus infects susceptible individuals who become infectious, and after some time they recover and become immune. When a vaccine exists and is applied to a susceptible individual, this also becomes a “recovered” [12].

The SIR model uses the strategy of compartments, related by a system of three differential equations. The initial value problem is presented in Eq. (1). The variables S , I and R represent, respectively, the number of susceptible, infected and recovered individuals. The term N represents the number of individuals, which is supposed to be constant, $S(t) + I(t) + R(t) = N$, $\forall t \geq 0$. The constant parameters are the transmission rate β , the recovery rate of infected individuals γ , and the birth/mortality rate μ . The measurement units of these parameters are the inverse of time unit. The birth rate and the mortality rate are assumed to be equal, in order to keep the population size constant. The values $1/\gamma$ and $1/\mu$ represent the average time of infection and the average lifetime of an individual, respectively, in time units (t.u.), w.l.g.. This model is realistic and mathematically and epidemiologically well-conditioned [12].

$$\begin{aligned} \frac{dS}{dt} &= \mu N - \mu S - \frac{\beta IS}{N}, & S(0) &\geq 0, \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I - \mu I, & I(0) &\geq 0, \\ \frac{dR}{dt} &= \gamma I - \mu R, & R(0) &\geq 0, \end{aligned} \quad (1)$$

By dividing Eq. (1) by N , considered constant, the system becomes expressed by the fraction of susceptible, s , and infected, i . This leads to Eq. (2). Therefore, the recovered fraction is computed as $r(t) = 1 - s(t) - i(t)$.

$$\begin{aligned} \frac{ds}{dt} &= \mu - \mu s - \beta is, & s(0) &\geq 0, \\ \frac{di}{dt} &= \beta is - \gamma i - \mu i, & i(0) &\geq 0, \end{aligned} \quad (2)$$

The basic reproduction number R_0 for the SIR model, defined in Eq. (3), represents the average number of infections produced by an infected individual. The SIR system has an asymptotically stable endemic equilibrium if and only if $R_0 > 1$, which is given by Eq. (4) [12].

$$R_0 = \frac{\beta}{\mu + \gamma} \quad (3)$$

$$s_e = \frac{1}{R_0}, \quad i_e = \frac{\mu(R_0 - 1)}{\beta} \quad (4)$$

The parameter values used here simulate a hypothetical disease, hard to be controlled, for which there exists an effective vaccine. It is supposed that the mean time for recovery is 7 t.u., which means that $\gamma = 1/7$ (t.u.)⁻¹. The considered population size is $N = 1000$ individuals. The “life expectancy” is 70 t.u., which implies $\mu = 1/70$ (t.u.)⁻¹, which may be interpreted as if a recovered individual would lose the immunity after that period. It is considered $\beta = 2.36$ (t.u.)⁻¹. Using Eq. (3), it comes an $R_0 = 15$, which is the basic reproduction number observed for the measles outbreak in England in 1856–1959 [2].

The initial condition is considered as $(s_0, i_0, r_0) = (0.99, 0.01, 0)$. Using Eq. (4), the endemic equilibrium point is $(s_e, i_e, r_e) \approx (0.067, 0.085, 0.848)$ and $(S_e, I_e, R_e) \approx (66.667, 84.848, 848.485)$. The number of infected individuals in the endemic equilibrium point is unacceptably high. For this reason, an optimal control is necessary to minimize the monetary costs to combat the disease and to minimize the infected population along time.

The time considered to determine the multiobjective control is $T_{ctrl} = T_{tmp} + T_{ss} = 150$ t.u., in which the first $T_{tmp} = 50$ t.u. are reserved to the transient control and next $T_{ss} = 100$ t.u. are reserved to steady-state control. The steady-state regime is supposed to be applied indefinitely, but here it will be evaluated just up to the time horizon T_{ss} .

The main idea of the transient control is to insert an aggressive set of vaccination campaigns with time intervals between them not necessarily equal and vaccination coverages also not constant. This set of interventions should be performed in order to avoid the big increase of the infected population in the initial stages of the epidemic process. This control should minimize the costs with the implementation of the campaigns and should also minimize the integral of the infected population during the time. It is defined here as a goal, in the simulations to be presented, that the proportion of infected population by the end of the transient control becomes less than or equal to 0.01. The aim of the steady-state control is to define a fixed time interval and a constant vaccination percentage for each campaign that must be implemented after the transient control. The values of time intervals between campaigns and vaccination percentages, in the steady-state regime, are designed in order to guarantee, in a theoretical way, that the number of infected individuals asymptotically tends to zero.

2.2. Individual Based Model

The Individual Based Model, or IBM, is known since the work by [13] and consists of a structure which simulates the relationships between individuals. A set of rules defines the possible interactions and how the individual characteristics may change. It allows the analysis of a system through the observation of the adaptive behavior properties of the individuals in a population. It is possible to extract information from IBM about variability among the individuals, local interactions, complete life cycles, and internal or external changes in the environment. In addition, it is easy to represent complex physical or social systems, although a big amount of data and computation may be necessary. This can reflect the

heterogeneity of the social interactions, allowing the representation of dynamic social networks.

This paper uses the IBM that was developed by [1], which is an improvement of the model proposed by [17]. This version was developed in order to reproduce the assumptions of the SIR model, with a stochastic model of interactions between individuals. Consequently the IBM converges, in average, to the SIR behavior.

Some epidemiological assumptions are adopted here. The parameters N , μ , β and γ have the same interpretations and values given in the SIR model in Section 2.1. The number of individuals is constant. Each individual assumes, on a time instant, a unique state related to its health-status. The mortality (and birth, consequently), recovery and vaccination are events that occur with uniform distribution in each discrete time interval. The event of “death” can be interpreted also as the loss of immunity of a recovered individual, which becomes susceptible again. On each discrete time instant, each individual can change the health category according to the following rules: (a) $S, I, R \rightarrow S$, an individual dies and another susceptible one is born; (b) $S \rightarrow I$, a susceptible individual gets contact with an infected individual and contracts the disease; (c) $I \rightarrow R$, the recovery of an infected individual occurs; (d) $S \rightarrow R$, the vaccination of a susceptible individual occurs.

The deterministic SIR model used in the optimization procedures considered here may be interpreted as a description of a situation in which a population is so large that the effect of stochastic variations in the events involving each individual becomes irrelevant in the total population. This paper uses an IBM in order to examine the behavior of the optimal policies that are synthesized by the multiobjective optimization procedure, in the stochastic environment of a finite set of individuals. Such kind of analysis is rather important in order to aid in the decision-making procedure that must be performed for choosing a single solution from the approximated Pareto-optimal set that is delivered from the multiobjective optimization procedure. The final set of non-dominated solutions from the optimization process is validated in a Monte Carlo simulation procedure which is performed using the IBM, producing the information that is used in an approximate stochastic nondominated sorting [16]. This step allows the identification of those solutions that would be sensitive to the stochastic environment, degenerating in poor performances. As a by-product, the probability of eradication and the probability of control of the disease are also obtained from the simulation with the IBM.

Fig. 1 illustrates the equivalence between the SIR model, based on a system of deterministic differential equations, and the IBM, based on stochastic difference equations, after a Monte Carlo simulation with 30 runs, using the above parameters and initial condition. The gray curves with different intensities represent the random behavior of the populations in different executions of the IBM.

3. Optimization models

In the problem formulation, $x \in X$ is the decision vector, X is the decision variable space, \mathcal{F}_x is the feasible set, $[f_1(x), \dots, f_m(x)]$ is the vector of objective functions to be minimized and \mathcal{X}^* is the set of efficient points, or Pareto-optimal set, which consists of all decision vectors for which the corresponding objective vector cannot be improved in any dimension without a degradation in another one. Given two decision vectors a and b , a is said to dominate b , ($a \prec b$), if and only if, for all $j \in \{1, \dots, m\}$ it is true that $f_j(a) \leq f_j(b)$ and there exists a $j \in \{1, \dots, m\}$ such that $f_j(a) < f_j(b)$. If it is not true that $a \prec b$ or $b \prec a$, it is said that a and b are mutually nondominated solutions. All decision vectors which are not dominated by

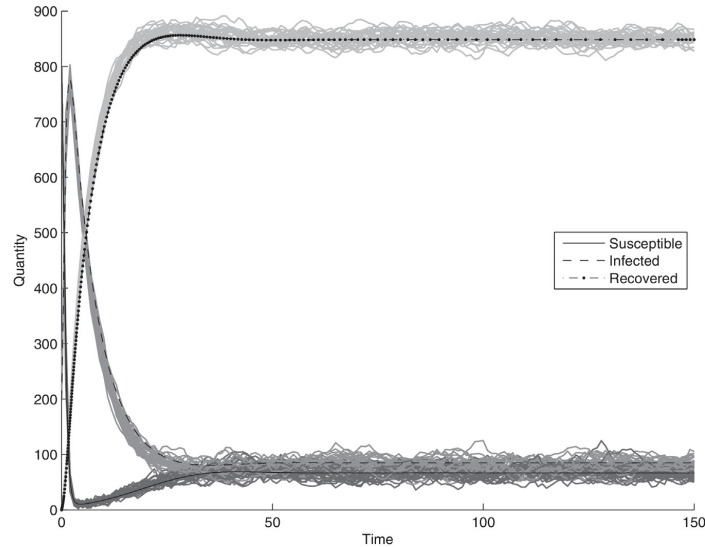


Fig. 1. IBM in 30 runs and the SIR model as the average behavior.

any other feasible decision vector are called nondominated solutions. Consequently, a Pareto-optimal solution is a vector x which is not dominated by any other vector of the feasible set \mathcal{F}_x , and the set of all Pareto-optimal solutions is the Pareto-optimal set λ^* . The image of the Pareto-optimal set in the objective function space is called the Pareto-front \mathcal{J}^* .

This paper proposes a multiobjective impulsive control using an open-loop continuous-variable dynamic optimization procedure to solve the problem of design of vaccination policies, using the SIR model as the dynamic system, coupled with a probabilistic and statistical analysis using the IBM. The main idea of impulsive control [22] is to split the continuous-time interval in some stages, performing impulsive actions of control just in some time instants. The dynamic system keeps its autonomous dynamics in the time intervals between the consecutive impulsive control actions. The problem of synthesis of impulsive control is sought here in an open-loop framework, using an evolutionary algorithm as the optimization machinery which allows rather arbitrary objective function and constraints.

The optimization process must determine the best control policies, minimizing the cost with vaccination campaigns and the integral of infected population over time, defining a bi-objective problem [6]. The optimization process is divided in two steps: the first one determines a set of nondominated steady-state control policies, and the second step determines a set of complete control policies, using a steady-state control policy selected from the set obtained previously, following a transient control which is determined in this step.

The multiobjective optimization model of the first step is described in Eq. (5). This first step intends to find a set of nondominated steady-state control regimes, $\lambda_{ss}^* = \{(\Delta\tau_{ss}, v_{ss})\}_{j=1}^{q_1}$, described by q_1 solutions to be determined with a fixed time interval between campaigns, $\Delta\tau_{ss}$, and with a fixed fraction of susceptible population which must be vaccinated in each campaign, v_{ss} . These are the decision variables. The steady-state regime should begin after the transient regime finishes, on the time T_{imp} , and should be applied indefinitely, up to the disease eradication.

The multiobjective optimization model of the second step is described in Eq. (7). The second step must determine a nondominated set of complete control regimes of vaccination. This step selects transient control policies that precede a steady-state control regime chosen from the set obtained previously. The set of nondominated complete control regimes of vaccination is described by

$$\lambda^* = \{(\Delta\tau_1, v_1, \Delta\tau_2, v_2, \dots, \Delta\tau_{M_{imp}}, v_{M_{imp}}, \Delta\tau_{ss}, v_{ss})\}_{j=1}^{q_2},$$

in which q_2 is the number of solutions. Each transient control is described by M_{imp} pairs of value of time interval between campaigns, $\Delta\tau_j$, and fraction of susceptibles to be vaccinated, v_j , $j \in \{1, 2, \dots, M\}$. The pair $(\Delta\tau_{ss}, v_{ss})$ is the selected steady-state regime that belongs to λ_{ss}^* .

The first objective function represents the integral of the infected population during the analyzed time interval of the steady-state control. The second objective represents the cost with the implementation of the campaigns during the steady-state control in the analyzed time horizon. The term M_{ss} is the number of campaigns, which is the floor of the ratio between the steady-state control time and the time interval value.

In the simulations to be presented here, it is assumed that there are fixed and variable costs involved in the implementation of vaccination campaigns. The fixed cost $c_1 = 10$ is assumed for each campaign. In this way, $c_1 \cdot M_{ss}$ is the fixed part of the cost with the implementation of all campaigns in the steady-state control. The term $c_2 \cdot M_{ss} \cdot (1 + v_{ss})^2$ describes the variable monetary cost related to the effort to reach v_{ss} percent of the susceptible population in all campaigns. In the simulations, it is assumed $c_2 = 1$. The term $c_3 \cdot \sum_{k=1}^{M_{ss}} v_{ss} \cdot S[k]$ is interpreted as the cost with all vaccines in all k vaccination campaigns, in which $c_3 = 1$ is assumed to be the

vaccine unitary cost. The resulting optimization problem for the steady-state phase, is described by (5).

$$\min_{\lambda_{ss}} \begin{cases} F_1 = \int_{T_{tmp}}^{T_{tmp}+T_{ss}} I(t) dt \\ F_2 = c_1 \cdot M_{ss} + c_2 \cdot M_{ss} \cdot (1 + v_{ss})^2 + c_3 \cdot \sum_{k=1}^{M_{ss}} v_{ss} \cdot S[k] \end{cases} \quad (5)$$

Subject to: { (6).

This problem has the following constraints, as described by (6). The system behavior is determined by a SIR model in the open time intervals when there is no vaccination. The initial condition is $s(0) = (\mu + \gamma) / \beta \approx 0.067$, which is the infection-free value of the SIR model under fixed pulse vaccination [20], and $i(0) = 0.01$, which is the maximum tolerable relative population size of infected individuals in the population. The k th campaign of vaccination occurs on each moment $\tau_k = T_{tmp} + k \cdot \Delta \tau_{ss}$, since the beginning of the steady-state control. On this moment an impulse will be provided (vaccination of the susceptibles), updating the population as described in $S(\tau_k^+)$, $I(\tau_k^+)$ and $R(\tau_k^+)$ equations. The notation $(\cdot)^+$ denotes the value of the variable just after the discontinuity caused by the impulse. The fraction of susceptible population to be vaccinated in each campaign must be in the interval $[v_{min}, v_{max}] = [0.40, 0.95]$. The time interval between each campaign must be greater than or equal to a minimum value, $\Delta t_{min} = \min \left(1, \left\lfloor \frac{v_{min} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{min}/2 - \gamma/\beta)} \right\rfloor \right)$, and less than or equal to a maximum value, $\Delta t_{max} = \max \left(20, \left\lfloor \frac{v_{max} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{max}/2 - \gamma/\beta)} \right\rfloor \right)$. This search range only includes values that theoretically guarantee that the infected population tends to zero [20]. Another constraint states that the disease should be under control during the steady-state control, meaning that the infected population should remain below an acceptable value, $i_{tol} = 0.01$.

$$\left\{ \begin{array}{l} \frac{dS}{dt} = \mu N - \mu S - \frac{\beta IS}{N}, \quad S(0) = s(0) \cdot N \geq 0; \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, \quad I(0) = i(0) \cdot N \geq 0; \\ \frac{dR}{dt} = \gamma I - \mu R, \quad R(0) = N - S(0) - R(0) \geq 0; t \in (\tau_k, \tau_{k+1}); \\ S(\tau_k^+) = S[k^+] = S[k](1 - v_{ss}); \\ I(\tau_k^+) = I[k^+] = I[k]; \\ R(\tau_k^+) = R[k^+] = R[k] + S[k]v_{ss}; \\ \tau_k = T_{tmp} + (k - 1) \cdot \Delta \tau_{ss}; \quad k = 1, \dots, M_{ss}; 0 < v_{min} \leq v_{ss} \leq v_{max} \leq 1.0; \\ \Delta t_{min} = \min \left(1, \left\lfloor \frac{v_{min} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{min}/2 - \gamma/\beta)} \right\rfloor \right); \\ \Delta t_{max} = \max \left(20, \left\lfloor \frac{v_{max} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{max}/2 - \gamma/\beta)} \right\rfloor \right); \quad 0 < \Delta t_{min} \leq \Delta \tau_{ss} \leq \Delta t_{max} \leq T_{ss}; \sum_{k=1}^{M_{ss}} \Delta \tau_{ss} \leq T_{ss}; \\ I(t) \leq i_{tol} \cdot N, \quad t \in (T_{tmp}, T_{tmp} + T_{ss}); \\ M_{ss} = \left\lfloor \frac{T_{ss}}{\Delta \tau_{ss}} \right\rfloor. \end{array} \right. \quad (6)$$

The control policy to be applied during the transient phase, as described by (7), should be time-variant. The objective functions of the second step of the design procedure, which perform the design

of that initial transient control, are similar to the ones employed in the first step, but now considering the whole time horizon.

$$\min_{\lambda^e} \begin{cases} F_1 = \int_0^{T_{tmp}+T_{ss}} I(t) dt \\ F_2 = c_1 \cdot M + c_2 \cdot \left(\sum_{k=1}^{M_{tmp}} (1 + v_k)^2 + M_{ss} \cdot (1 + v_{ss})^2 \right) \\ \quad + c_3 \cdot \sum_{k=1}^M v_k \cdot S[k] \end{cases} \quad (7)$$

Subject to: { (8).

The optimization problem in this second step has more constraints than in the first step, as described by (8). The system is still dictated by the SIR model with the same parameters as before. The initial condition describes a bad situation which must be controlled. The initial relative sizes of susceptible, infected and recovered populations are assumed to be $s(0) = 0.80$, $i(0) = 0.20$ and $r(0) = 0$, respectively, in the simulations presented here. Each campaign k is described by the pair (τ_k, v_k) , representing the fraction of susceptible population to be vaccinated and the time of vaccination. In the transient regime, the time interval $\Delta \tau_k$ and the vaccination coverage, v_k , are time-variant. The sum of time intervals obey their limits T_{tmp} and T_{ss} . The disease must be under control by the beginning of steady-state control, meaning that the number infected individuals must be less than or equal to a tolerable value $i_{tol} \cdot N$. The value $i_{tol} = 0.01$ is assumed in the simulations presented here. The number of campaigns, M_{tmp} , in transient control can vary from $\lceil T_{tmp} / \Delta t_{max} \rceil$ and $\lfloor T_{tmp} / \Delta t_{min} \rfloor$. The number of analyzed steady-state control stages, M_{ss} is the same as before. The total number of campaigns is $M = M_{tmp} + M_{ss}$.

$$\left\{ \begin{array}{l}
\frac{dS}{dt} = \mu N - \mu S - \frac{\beta IS}{N}, \\
\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, \\
\frac{dR}{dt} = \gamma I - \mu R, \\
S(\tau_k^+) = S[k^+] = S[k](1 - v_k); \\
I(\tau_k^+) = I[k^+] = I[k]; \\
R(\tau_k^+) = R[k^+] = R[k] + S[k]v_k; \\
\tau_k = \tau_{k-1} + \Delta \tau_k; \\
v_k = \begin{cases} v_k, & k \leq M_{tmp}, \\ v_{ss}, & M_{tmp} + 1 \leq k \leq M_{ss}; \end{cases} \\
\Delta \tau_k = \begin{cases} \Delta \tau_k, & 1 \leq k \leq M_{tmp}, \\ \Delta \tau_{ss}, & M_{tmp} + 1 \leq k \leq M_{ss}; \end{cases} \\
\Delta t_{min} = \min \left(1, \left\lfloor \frac{v_{min} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{min}/2 - \gamma/\beta)} \right\rfloor \right); \\
\Delta t_{max} = \max \left(20, \left\lfloor \frac{v_{max} \cdot \gamma}{\beta \cdot \mu \cdot (1 - v_{max}/2 - \gamma/\beta)} \right\rfloor \right); \\
I(t) \leq i_{tol} \cdot N, \\
M_{tmp} \in \{ \lfloor T_{tmp}/\Delta t_{max} \rfloor, \lfloor T_{tmp}/\Delta t_{max} \rfloor + 1, \lfloor T_{tmp}/\Delta t_{max} \rfloor + 2, \dots, \lfloor T_{tmp}/\Delta t_{min} \rfloor \}; \\
M_{ss} = \left\lfloor \frac{T_{ss}}{\Delta t_{ss}} \right\rfloor; \\
M = M_{tmp} + M_{ss}.
\end{array} \right. \quad \begin{array}{l}
S(0) = s(0) \cdot N \geq 0; \\
I(0) = i(0) \cdot N \geq 0; \\
R(0) = N - S(0) - R(0) \geq 0; t \in (\tau_k, \tau_{k+1}); \\
k = 1, \dots, M; 0 < v_{min} \leq v_k \leq v_{max} \leq 1.0; \\
0 < \Delta t_{min} \leq \Delta \tau_k \leq \Delta t_{max} \leq T_{ss}; \sum_{k=1}^{M_{tmp}} \Delta \tau_k \leq T_{tmp}; \sum_{k=1}^{M_{ss}} \Delta \tau_{ss} \leq T_{ss}; \\
t \in (T_{tmp}, T_{tmp} + T_{ss});
\end{array} \quad (8)$$

Deterministic optimization techniques are not good choices to solve those types of problems, because they are nonlinear with many local minima. Moreover, the decision variables can have different feasible ranges depending on the evaluation of the system states. In this context, a better choice would be the use of stochastic search algorithms.

The next section explains the optimization engine used in this work, which is a multiobjective evolutionary algorithm with two local search mechanisms. One local search procedure is based on the single-objective optimization of convex quadratic approximations of the objective functions. The other local search procedure uses a hash table to store all solutions evaluated so far. This memory is used in order to avoid the reevaluation of repeated solutions. When an attempt is made to evaluate a repeated solution, a new solution which has not been evaluated before is generated in its neighborhood.

4. Optimization engine: a modified NSGA-II

Evolutionary multiobjective optimization algorithms are computational tools for solving problems with multiple conflicting objective functions which can deal with difficulties such as discontinuity, non-convexity, multimodality, presence of noise and so forth [10]. This work employs a variation of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) presented in [9] to determine the nondominated solutions.¹

¹ The NSGA-II performs the fitness assignment according to the dominance level of each individual. For example, the individuals that are dominated by no other solution receive the frontier index 1, the ones that are not dominated by any other solution except by solutions in frontier 1 receive index 2, those that are not dominated by any other solution except by solutions in frontiers 1 and 2 receive index 3, and so

4.1. Local search via convex quadratic approximation

Deterministic search methods have fast and precise convergence properties, but they are limited to find only local optima. The hybridization between evolutionary and deterministic search methods, in order to enhance local search, has arisen as a promising idea. This work applies a local search scheme proposed in [7,8] in order to speed up the convergence and enhance the quality of solutions. The hybrid method picks up points from the non-dominated population and determines, via linear programming, a convex quadratic approximation with diagonal Hessian for each objective function. Since these quadratic approximations of the functions and the respective weighted sums are convex, fast deterministic methods can be used in order to generate approximated Pareto-optimal solutions from the approximated functions.

The hybrid algorithm is constructed taking the basic NSGA-II algorithm in combination with a local search which is executed at each 20 generations. The local search performs the following actions: (1) A set of r points is chosen at random from the current nondominated solution set. (2) A set of m new points is generated randomly, with Gaussian distribution, with standard deviation equal to 0.01 times the size of the search space for each dimension, around each point chosen in the former step. (3) All objective functions are evaluated on the new points. (4) A convex quadratic approximation is calculated for each objective function, for each one of the r sets of $m + 1$ points. (5) The quadratic

forth. An efficient version of this procedure, proposed by [14], is employed in this work.

approximation functions are used in order to generate p estimates of Pareto-optimal points, using a weighted-sum (convex combination) procedure and a quadratic programming algorithm. (6) The new $r \times p$ points are merged to the current population of solutions in NSGA-II.

In the experiments conducted here, $r=4$ solutions are selected at random from the current nondominated set, and $m=2s$, with $s=2 \cdot n + 1$, in which n is dimension of the problem. The number of points generated for each set of approximated functions has been chosen as $p=N/2$, in which N is the population size of the evolutionary algorithm. This means that, in each generation when the local search is conducted, $2N$ new points are generated, and the selection procedure is performed over $4N$ individuals, with $2N$ individuals coming from the usual iteration of NSGA-II and $2N$ coming from the local search. The convex combinations are performed using random positive weights with uniform distribution.

4.2. Local search using hash table

In order to avoid the re-evaluation of solutions, a hash table was incorporated in the evolutionary algorithm. A hash table is a data structure which uses a hash function to map key values to their associated values. The hash function transforms the key into the index which locates the position where the corresponding value is stored. When the number of keys actually stored is small in relation to the total number of possible keys, hash tables become an effective alternative to directly addressing an array, since a hash table typically uses an array of size proportional to the number of keys actually stored [5]. The average time to search for an element in a hash table, with a good data structure and a good hash function, is $O(1)$ [21].

The present work uses a hash table in order to store the solutions obtained during the evolutionary iterations in the optimization process. When a new individual is generated by a genetic operator it is verified whether this is a preexisting solution. If there is a repetition in the hash table, a random vector with Gaussian distribution, zero-mean and standard deviation equal to 1.0 percent of the search space dimension is added to the individual coordinates, generating a new individual. In this way, only new solutions are evaluated. In the local search phase, when a repeated solution is generated, just a copy of the objective function is retrieved in order to generate the quadratic approximation.

As explained in the next subsection, the optimization algorithm is executed many times and on each run the hash table is updated. In this way, the database of solutions begins empty and increases in each execution of the NSGA-II. On the end of all executions, a nondominated sorting is applied over all stored solutions. The final result is a set with all nondominated solutions which were evaluated during the optimization process.

5. Experiment design

The pseudocode which solves the two phases of the optimization problem is depicted in Algorithm 1. First, a random population of solutions, P_1 , is generated. Then, the solutions are analysed in the hash table, H , changing the repeated solutions for other ones in their neighbourhood. The parent population is evaluated and stored in the hash table. In the next step, all solutions are classified according to their front level and receive their respective crowding distance level. In each generation loop, g , selection and recombination are executed with the parent population, P_g , generating the offspring population Q_g , which may suffer mutation in the next step. After, the offspring population is analysed and modified, whether necessary, according to the hash table of solutions. If it is time to run local search, a subset of solutions is selected at random, beginning

from the first level of nondominated set of the current generation to execute the procedure, and the returned solutions are added to the offspring population. The offspring solutions are evaluated and inserted in the hash table. The parent and offspring populations are merged and classified according to dominance relation level. The last step of the iteration selects the best solutions to become the parents in the next generation.

Algorithm 1. Pseudo-code for NSGA-II with local search and hash table.

```

Input:  $H$ 
Output:  $H, \mathcal{X}$ 
 $P_1 \leftarrow \text{InitialPopulation}()$ 
 $[H, P_1] \leftarrow \text{AnalyseInsertHashTable}(H, P_1)$ 
 $[H, P_1] \leftarrow \text{EvaluatePopulation}(H, P_1)$ 
 $\lambda^{(1)} \leftarrow \text{NonDominatedSorting}(P_1)$ 
 $\lambda^{(1)} \leftarrow \text{CrowdingDistance}(\lambda^{(1)})$ 
for  $g$  from 1 to  $ng$  step 1 do
   $P_g \leftarrow \text{SelectionAndRecombination}(P_g)$ 
   $Q_g \leftarrow \text{Mutation}(Q_g)$ 
   $[H, Q_g] \leftarrow \text{AnalyseInsertHashTable}(H, Q_g)$ 
  if it is desired to execute local search then
     $L \leftarrow \text{LocalSearch}(\lambda^{(g)}, H)$ 
     $Q_g \leftarrow Q_g \cup L$ 
  end if
   $[H, Q_g] \leftarrow \text{EvaluatePopulation}(H, Q_g)$ 
   $R_g \leftarrow P_g \cup Q_g$ 
   $\lambda^{(g)} \leftarrow \text{NonDominatedSorting}(R_g)$ 
   $P_{g+1} \leftarrow \text{FillNondominated}(R_g, \lambda^{(g)})$ 
end for

```

The values of algorithm parameters were chosen after a large number of computational experiments. The number of generations was set to 50, the population size to 40, the probability of recombination to 1, the probability of mutation to 0.2 and the distribution index for crossover and mutation to 10. For each step of the optimization process the algorithm was run 10 times. Initially, an empty hash table is defined, being updated along each execution of the evolutionary algorithm. In the end, a nondominated sorting procedure is executed over all solutions in the hash table.

After the end of the first optimization step, one nondominated solution among all solutions is selected to represent the steady-state control policy which will be adopted. The chosen steady-state regime solution is the nearest one in relation to Utopian (or ideal) solution according to the Euclidean distance. Using the selected steady-state regime, the second step of the optimization is executed in order to determine a set of transient regimes. In the end of all executions of the evolutionary algorithm, a nondominated sorting procedure is performed over all solutions in the hash table. The result is a global nondominated set with complete control policies, including transient and steady-state stages.

The solutions in the global nondominated set with complete control policies are evaluated using an IBM. A Monte Carlo procedure with 1000 runs is performed for each complete vaccination control policy in order to analyse the behavior of the epidemics in the presence of stochasticity. The vaccination percentage is now interpreted as the probability of a susceptible individual being vaccinated. In each simulation, the objective functions are evaluated and stored for an *a posteriori* analysis.

An approximated stochastic dominance analysis is performed in the solutions obtained by the optimization process by using the IBM simulation. This step is performed in order to eliminate the non-robust solutions, which suffer degradation under the stochastic environment. The new nondominated set is determined using the quantiles 0.25, 0.50 and 0.75 of the re-evaluated objective functions. Only robust solutions, which remain non-dominated in the stochastic simulation, are kept.

Using the solutions in the stochastically nondominated set, the following two frequencies are computed: the number of realizations in which the disease is eradicated, and the number of

realizations in which the infected population remains below the tolerable value from the end of the transient control until the end of the time horizon of the steady-state control.

6. Results

After the execution of the multiobjective optimization algorithm for the first step of the problem, a set of nondominated steady-state solutions was determined, with 159 solutions. All those solutions satisfy all constraints. In all runs in this first step, the evolutionary algorithm evaluated 23645 different solutions.

The image of the feasible solutions are illustrated in Fig. 2. The black points are the nondominated solutions, the red solutions are those which were generated at random in the first population of parents, the green solutions are those generated by genetic operators of recombination and mutation, the yellow solutions were generated by the local search operator based on a hash table, the light blue are solutions generated randomly in order to generate the convex quadratic approximation, and the blue ones are those that correspond to optima returned by local search. Observe the importance of the use of hash table and of the local search to steer the convergence to the nondominated set.

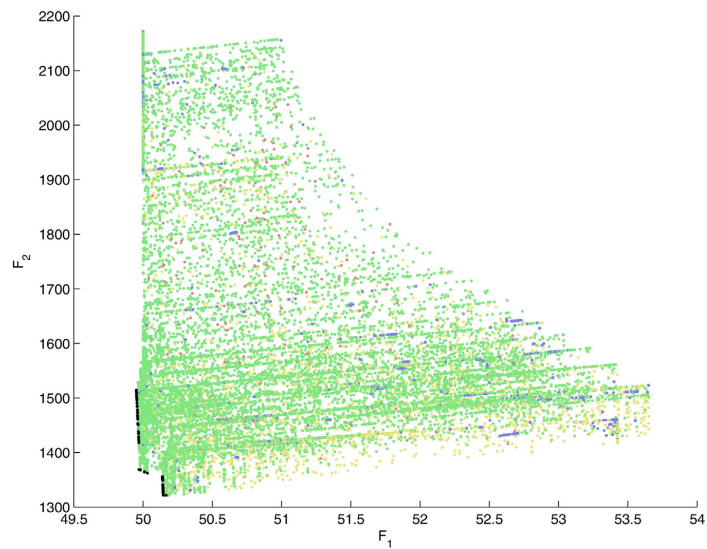


Fig. 2. Image set of all feasible solutions evaluated in all executions of the evolutionary algorithm, in all runs for the problem presented in Eq. (5).

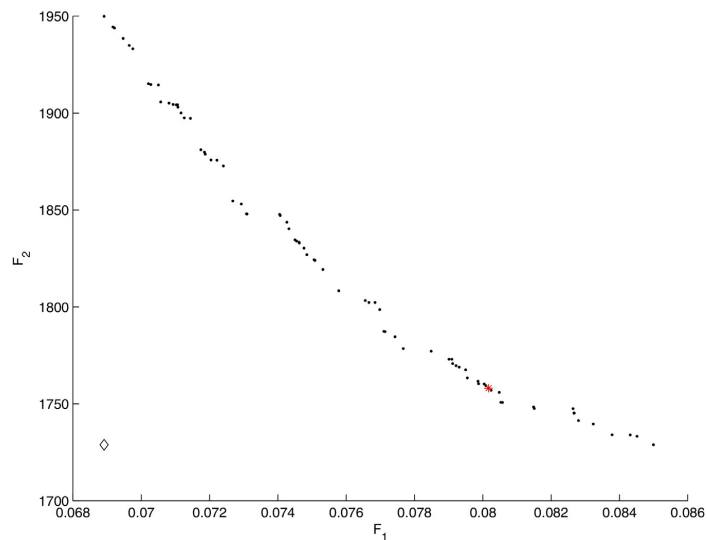


Fig. 3. Nondominated front of the problem presented in Eq. (7) considering all executions of the evolutionary algorithm.

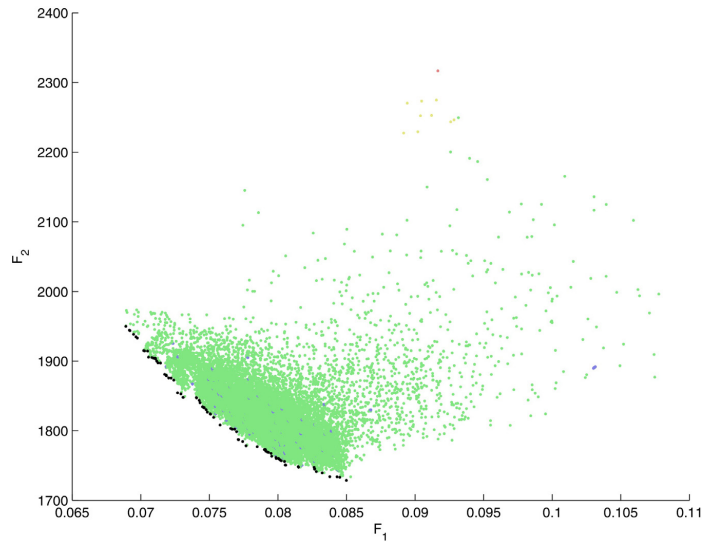


Fig. 4. Image set of all feasible solutions evaluated in all executions of the evolutionary algorithm in all runs for the problem presented in Eq. (7).

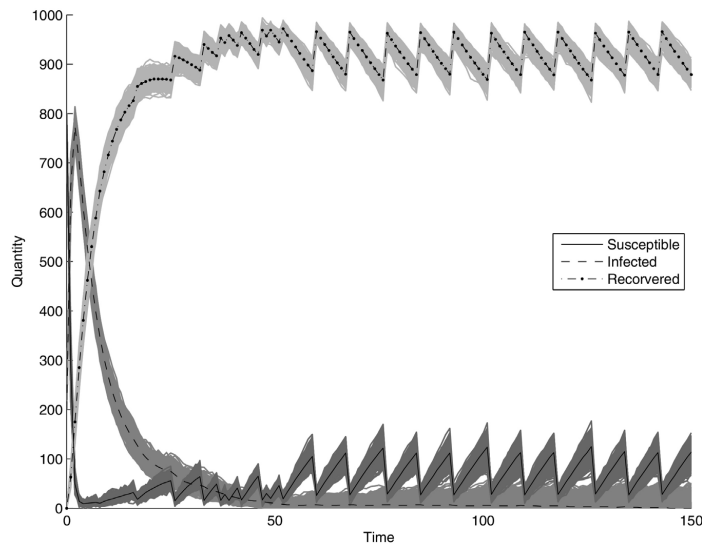


Fig. 5. Behavior of solution 25, from the stochastically nondominated set presented in Table 1.

Before running the second step, the steady-state control policy was chosen selecting the Utopian nearest nondominated solution of the first step. This corresponds to the steady-state policy $\Delta \tau_{ss} = 8.3335$ and $\nu_{ss} = 0.8752$ which was included in Eq. (7).

The Pareto front of the second step of the optimization process is illustrated in Fig. 3. All those solutions satisfy all constraints. The diamond suit is the Utopian solution. The final nondominated set of complete controls has 96 solutions. These solutions are evaluated using an IBM.

The image of all 127980 solutions evaluated in all executions of the second step is illustrated in Fig. 4. The colors of the

solutions have the same interpretation as in Fig. 2. The number of repeated solutions was not as big as in the first step. Clearly, the generation of solutions in the neighborhood of the selected point to apply local search was important to improve the solutions. Similarly, the optimization of the sum of the convex quadratic approximations to obtain better solutions on each local search step was important. Finally, the usage of the hash table to store and generate/evaluate only new solutions improves the variability.

As mentioned before, all solutions in the final nondominated set of complete control policies were simulated 1,000 times in the

Table 1
Final set of robust solutions determined by stochastic dominance.

#	F_1			F_2			P_c	P_e
	$Q_{0.25}$	$Q_{0.50}$	$Q_{0.75}$	$Q_{0.25}$	$Q_{0.50}$	$Q_{0.75}$		
1	50.5232	53.1987	58.6689	1832.6193	1950.6193	2011.6193	0.7410	0.5910
2	50.7616	53.2781	58.9338	1834.2693	1950.2693	2007.2693	0.7452	0.5910
3	50.9404	53.9272	59.1457	1814.9347	1929.9347	1997.9347	0.7213	0.5790
4	50.9868	54.0596	59.6093	1804.8550	1924.8550	1997.8550	0.7157	0.5540
5	51.0397	53.4503	58.5563	1855.6230	1974.6230	2031.6230	0.7631	0.6230
6	51.0993	53.4238	58.2384	1810.5349	1921.5349	1969.5349	0.7885	0.6360
7	51.2384	53.3046	58.2715	1825.8844	1933.8844	1985.8844	0.7964	0.6410
8	51.2715	54.2450	59.7219	1806.9604	1925.9604	1990.9604	0.7172	0.5590
9	51.2848	53.3444	58.1589	1800.8713	1908.8713	1953.8713	0.8011	0.6330
10	51.5033	53.8477	58.6755	1785.6212	1897.6212	1950.6212	0.7846	0.6360
11	51.7483	54.2384	58.6755	1766.2756	1866.2756	1920.2756	0.7879	0.6320
12	51.7881	53.7417	58.8146	1788.4076	1895.4076	1942.4076	0.7926	0.6460
13	51.7881	54.0331	58.6225	1795.1285	1894.1285	1946.1285	0.7942	0.6460
14	51.8940	54.3907	59.2053	1761.4848	1868.4848	1926.4848	0.7722	0.5990
15	51.9735	54.9404	60.1656	1756.7363	1869.7363	1931.7363	0.7310	0.5650
16	52.0331	54.7219	60.0397	1747.5504	1856.5504	1920.5504	0.7465	0.5850
17	52.4040	55.5099	60.6755	1747.2045	1856.2045	1928.2045	0.7151	0.5590
18	52.4901	55.6026	60.8212	1728.3113	1837.3113	1905.3113	0.7310	0.5670
19	52.7020	55.6954	60.4702	1717.7163	1829.7163	1895.7163	0.7282	0.5600
20	53.2583	57.1258	62.0596	1675.8660	1788.8660	1865.8660	0.6805	0.5200
21	53.4437	56.6755	61.4437	1672.2291	1772.2291	1839.2291	0.7129	0.5450
22	53.5563	57.3907	62.1788	1656.5775	1753.5775	1834.5775	0.6638	0.5010
23	53.6026	57.0066	62.3444	1654.0031	1762.0031	1838.0031	0.6692	0.5130
24	53.6291	57.2252	62.1523	1668.1927	1767.1927	1848.1927	0.6695	0.5060
25	53.6954	57.5099	62.5762	1641.6787	1740.6787	1823.6787	0.6658	0.5110
26	53.7947	57.8212	62.5695	1627.6702	1721.6702	1807.6702	0.6455	0.4890
27	54.3444	59.0596	63.4570	1617.9675	1713.9675	1818.9675	0.5974	0.4410
28	54.9735	59.8742	64.2781	1606.2816	1692.2816	1795.2816	0.5461	0.3970
29	54.9868	59.6026	64.0927	1605.4870	1695.4870	1802.4870	0.5625	0.4130
30	55.2848	59.9735	64.2185	1607.0324	1693.0324	1788.0324	0.5409	0.3960

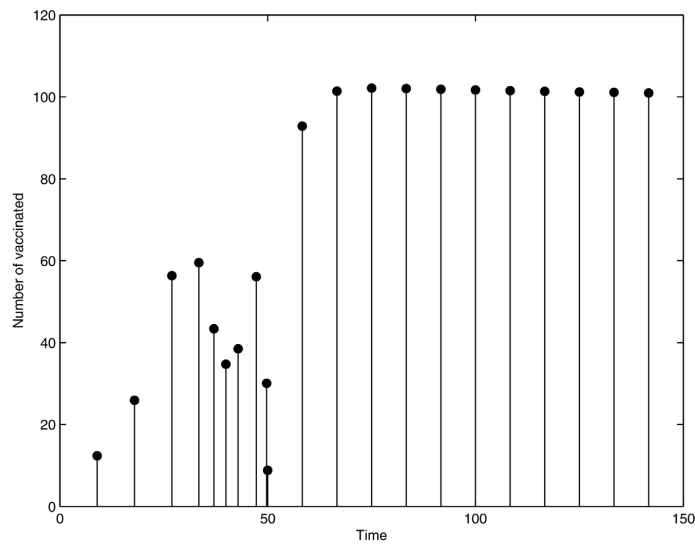


Fig. 6. Mean number of vaccinations for the solution 25 from the stochastically nondominated set presented in Table 1.

IBM. A stochastic nondominated sorting procedure was executed using the re-evaluations of the objective functions. The reference values for defining the stochastic dominance were the quantiles 0.25, 0.50 (the median) and 0.75 of the two objectives. The stochastically nondominated set of robust complete control policies is presented in Table 1. This table shows the objective function quantiles, the probability, P_c , that the disease is “under control” after

time 50 t.u., defined as the proportion of realizations in which the number of infected remained under 0.01N, and the probability of disease eradication, P_e , defined as the proportion of realizations in which the infected population reaches zero within the simulation time horizon.

Note that just 30 solutions were found to be robust, from the 96 ones that were in the nondominated set obtained after the

optimization procedure. The joint variation of F_1 and F_2 in the nondominated solutions provide the information about the existing trade-off between objectives. The values of P_c and P_e indicate, respectively, good chances to keep under control or to eradicate this disease.

This stochastically nondominated set is the final outcome delivered by the procedure proposed here. One control policy must be chosen by the decision-maker, in order to be applied in practice. Suppose that the highlighted solution in Fig. 3, presented in Table 1 with index 25, is selected. The stochastic behavior of this solution is illustrated in Fig. 5. The black curves represent the median behavior

over the time. The mean number of vaccinated in each campaign is presented in Fig. 6.

Finally, a comparison between the proposed control policy (PCP) and another strategy is presented. The methodology used for comparison is represented by a Steady-State Control (SSC) solution, in which the vaccination rate and time interval between campaigns are always constants. In this case, it will be used the steady-state control with $v_{ss} = 0.8595$ and $\Delta\tau_{ss} = 7.1430$ t.u., which was determined by the first step of the proposed methodology, according to Eq. (5). The stochastic behavior of SSC is illustrated by Fig. 7 and its number of vaccinated individuals is shown in Fig. 8.

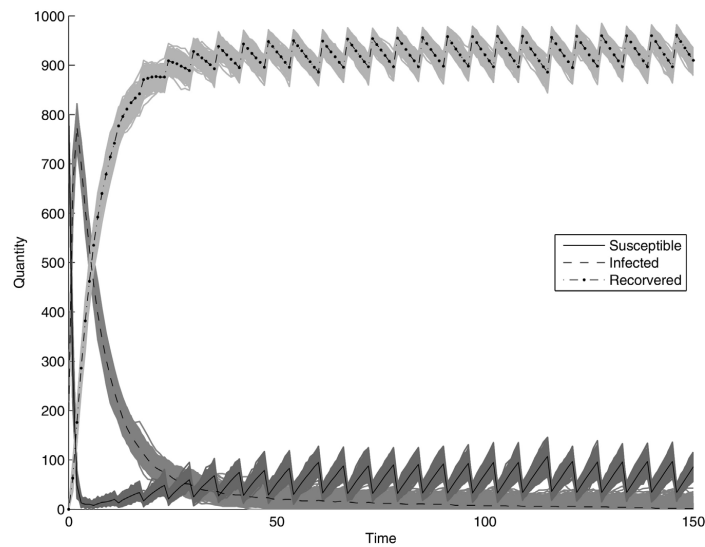


Fig. 7. Behavior of the steady-state control (SSC) with vaccination rate $v_{ss} = 0.8595$, and time interval $\Delta\tau_{ss} = 7.1430$ t.u.

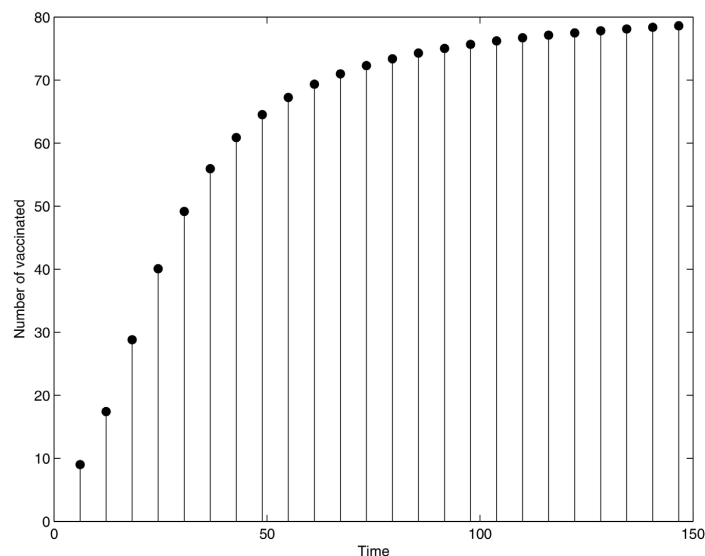


Fig. 8. Mean number of vaccinations for the steady-state control (SSC).

Table 2

Performances of (PCP) the proposed control policy (the robust solution selected from row 25 of Table 1), and (SSC) the steady-state control.

#	F_1			F_2			P_c	P_e
	$Q_{0.25}$	$Q_{0.50}$	$Q_{0.75}$	$Q_{0.25}$	$Q_{0.50}$	$Q_{0.75}$		
PCP	53.6954	57.5099	62.5762	1641.6787	1740.6787	1823.6787	0.6658	0.5110
SSC	56.1457	59.5232	62.8079	1688.6331	1746.6331	1816.6331	0.5607	0.4840

The performance indices of the proposed control policy (PCP), corresponding to the control policy indicated in row 25 of Table 1, and the steady-state control (SSC) policy are presented in Table 2.

The data presented in Table 2 and Figs. 5–8 are observed now, for the sake of comparison. It becomes clear that the proposed control is better than the steady-state control in almost all performance indices, except one (the $Q_{0.75}$ percentile of function F_2), meaning

that it tends to be both financially cheaper and more effective in the disease control. It is noticeable that the probability of disease eradication is increased from 48% to 51% (an increase of more than 6%) when the proposed control is employed, which is much more significant than the median enhancement of nearly 1.6% in F_1 and of nearly 0.1% in F_2 . This can be explained by the “strategic policy” that can be synthesized for the transient phase of control in

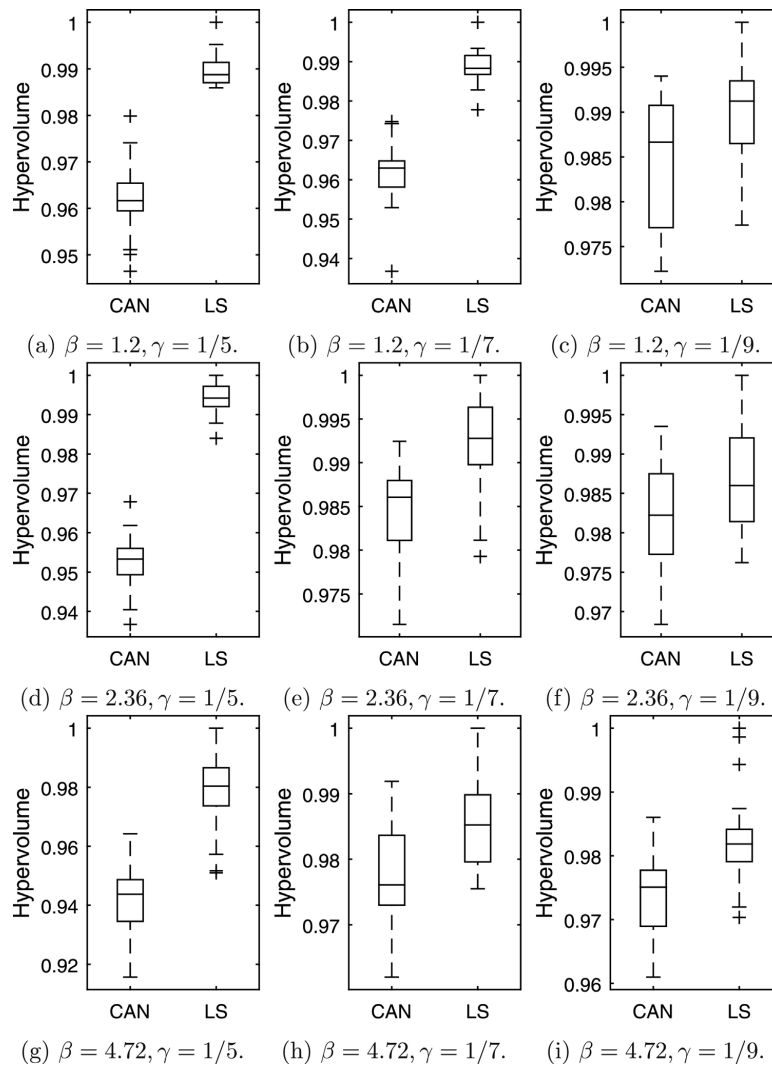


Fig. 9. Normalized hypervolumes for each experiment, considering different values of β and γ .

Table 3
P-values returned by Wilcoxon paired test for each pair of β and γ .

β	γ		
	1/5	1/7	1/9
1.18	0.0000	0.0000	0.0032
2.36	0.0000	0.0000	0.0436
4.72	0.0000	0.0002	0.0000

the proposed methodology. Notice that the PCP solution provides a more frequent vaccination control than the SSC control on the times just after the peak of infection, providing a period of time in which the number of susceptibles becomes very low and the number of recovered becomes very high – due to the combined effect of the vaccination and the previous infection peak. This leads to an enhanced number of realizations in which the disease is eradicated. Such kind of “strategic policy” cannot be synthesized by a steady-state control.

7. Optimization engine comparison

Additionally to the previous case study, a test was performed in order to evaluate the effectiveness of the proposed optimization engine with local search (LS) in comparison with the canonical NSGA-II (CAN).

A total of nine computational experiments were planned and executed considering different disease parameters. For each pair of $\gamma \in \{1/5, 1/7, 1/9\}$ and $\beta \in \{1.18, 2.36, 4.72\}$, the versions CAN and LS were run independently 30 times in each phase of the optimization problem. This was done with the same number of objective function evaluations, 1000, for each version and optimization phase, in order to obtain the respective set of nondominated complete control policies. The other parameter values were kept as presented before. For each solution set, the hypervolume indicator quality metric [35] was calculated using a suitable fixed reference point. The greater the hypervolume value, the best is the returned set of solutions.

In this way, for each experiment the 30 hypervolumes were compared using the nonparametric Wilcoxon paired test [36]. The null hypothesis, which states that the difference between the two hypervolume medians are equal, is analyzed with confidence 0.95. This means that if the statistical test returns a p -value less than 0.05 then the hypervolumes may be considered different.

The normalized hypervolume data, for each experiment, are presented in Fig. 9. Statistical differences were detected in all experiments according to the p -values presented in Table 3. Therefore, it can be concluded that the proposed optimization engine with local search was superior in all experiments considering diseases with different dynamical parameters.

8. Conclusion

This paper presented a multiobjective dynamic optimization methodology for the synthesis of a set of control policies that minimize the integral of the infected population and the cost of the implementation of vaccination campaigns. The synthesis, in both cases of the transient control and of the steady-state control, is performed using a SIR model of the epidemics. To solve the synthesis problem, an evolutionary algorithm is designed, which provides efficient solutions in a reasonable computational time.

Some features of the resulting solutions are:

- The transient regime of the infection, which occurs immediately after the disease outbreak in population, is tackled via a transient vaccination policy which considers both the time interval between vaccination campaigns and the vaccination effort in each

campaign as decision variables. In this way, the usual large peaks of infection are mitigated at an affordable cost.

- As the infection reaches an endemic stage, the control policy becomes formulated as a steady-state periodic control, with fixed time interval between campaigns and fixed vaccination effort in each campaign. Those parameters are defined in such a way that the epidemics will be eventually eradicated. This steady-state control policy is concatenated with the transient control policy, in this way defining a complete control policy.

A whole set of Pareto-optimal solutions of the complete control problem is determined via multiobjective algorithm, representing different trade-offs between monetary cost and effectiveness. After the set of Pareto-optimal control policies is determined, the solutions belonging to this set are submitted to Monte Carlo simulations, in order to evaluate the magnitude of the effect of stochasticity associated to finite-size populations. A stochastic dominance analysis is performed, and the dominated solutions are excluded. As a by-product, the Monte Carlo analysis provides an estimate of the probability of eradication of the disease within a given time horizon. In this way the proposed approach delivers multiple robust solutions for the problem of vaccination policy synthesis which can be considered by a decision-maker.

Some limitations of the models employed here consist, for instance, in lack of age structure, spatial structure and seasonality, both in the SIR model and in the IBM. However, it should be noticed that the proposed approach of dividing the synthesis problem in two phases – the steady-state and the transient ones – and the robustness analysis using a stochastic model and a stochastic dominance approach can be transposed to other situations, using models that are more detailed and realistic. Future works will consider real-life applications.

Acknowledgements

The authors acknowledge the support by the Brazilian agencies Capes, CNPq and Fapemig.

References

- [1] A.C.L. Almeida, Modelos Matemáticos de Propagação de Epidemias Baseados em Redes Sociais e Detecção de Clusters de Doenças (PhD thesis), Universidade Federal de Minas Gerais, 2011 (in Portuguese).
- [2] R.M. Anderson, R.M. May, Infectious Diseases of Humans: Dynamics and Control, Oxford University Press, USA, 1992.
- [3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd ed., MIT Press, 2009.
- [4] A.R. da Cruz, R.T.N. Cardoso, R.H.C. Takahashi, Multi-objective design with a stochastic validation of vaccination campaigns, in: Preprints of the IFAC Workshop on Control Applications of Optimisation (CAO'09), vol. 7 (1), 2009, pp. 1–6.
- [5] A.R. da Cruz, E.F. Wanner, R.T.N. Cardoso, R.H.C. Takahashi, Using convex quadratic approximation as a local search operator in evolutionary multiobjective algorithms, in: IEEE Congress on Evolutionary Computation (CEC), IEEE, 2011, pp. 1217–1224.
- [6] A.R. da Cruz, R.T.N. Cardoso, R.H.C. Takahashi, Multiobjective dynamic optimization of vaccination campaigns using convex quadratic approximation local search, in: Evolutionary Multi-Criterion Optimization, EMO 2011, Springer, 2011, pp. 404–417.
- [7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
- [8] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1995) 1–16.
- [9] V. Grimm, S.F. Railsback, Individual-Based Modeling and Ecology, Princeton University Press, 2005.
- [10] H.W. Hethcote, The mathematics of infectious diseases, SIAM Rev. 42 (4) (2000) 599–653.
- [11] M. Huston, D. DeAngelis, W. Post, New computer models unify ecological theory, Bioscience 38 (10) (1988) 682–691.
- [12] M.T. Jensen, Reducing the run-time complexity of multiobjective EAS: the NSGA-II and other algorithms, IEEE Trans. Evol. Comput. 7 (5) (2003) 503–515.

- [15] W.O. Kermack, A.G. McKendrick, Contributions to the mathematical theory of epidemics. III. Further studies of the problem of endemicity, *Proc. R. Soc. Lond. Ser. A* 141 (843) (1933) 94–122.
- [16] H. Levy, Stochastic dominance and expected utility: survey and analysis, *Manag. Sci.* (1992) 555–593.
- [17] E.G. Nepomuceno, R.H.C. Takahashi, L.A. Aguirre, Individual Based-Model (IBM): an alternative framework for epidemiological compartment models, *Revista Brasileira de Biometria* 34 (2016) 133–162.
- [20] B. Shulgin, L. Stone, Z. Agur, Pulse vaccination strategy in the sir epidemic model, *Bull. Math. Biol.* 60 (6) (1998) 1123–1148.
- [21] T.G. Szymanski, Hash table reorganization, *J. Algorithms* 6 (3) (1985) 322–335.
- [22] T. Yang, Impulsive control, *IEEE Trans. Autom. Control* 44 (5) (2002) 1081–1083.
- [23] A. Hinman, Eradication of vaccine-preventable diseases, *Ann. Rev. Public Health* 20 (1) (1999) 211–229.
- [24] R. Kim-Farley, Global immunization, *Ann. Rev. Public Health* 13 (1) (1992) 223–237.
- [25] X. Yan, Y. Zou, Optimal and sub-optimal quarantine and isolation control in SARS epidemics, *Math. Comput. Model.* 47 (1) (2008) 235–245.
- [26] X.M. Hu, J. Zhang, H. Chen, Optimal vaccine distribution strategy for different age groups of population: a differential evolution algorithm approach, *Math. Probl. Eng.* (2014).
- [27] W.O. Dias, E.F. Wanner, R.T. Cardoso, A multiobjective optimization approach for combating *Aedes aegypti* using chemical and biological alternated step-size control, *Math. Biosci.* 269 (2015) 37–47.
- [28] G. Zaman, Y.H. Kang, I.H. Jung, Stability analysis and optimal vaccination of an SIR epidemic model, *Biosystems* 93 (3) (2008) 240–249.
- [29] I.M. Longini, E. Ackerman, L.R. Elveback, An optimization model for influenza A epidemics, *Math. Biosci.* 38 (1978) 141–157.
- [30] A.N. Hill, I.M. Longini, The critical vaccination fraction for heterogeneous epidemic models, *Math. Biosci.* 181 (1) (2003) 85–106.
- [31] J.T. Wu, M. Lawrence, A.S. Perelson, Optimization of influenza vaccine selection, *Oper. Res.* 53 (3) (2005) 456–476.
- [32] R. Patel, I.M. Longini, M.E. Halloran, Finding optimal vaccination strategies for pandemic influenza using genetic algorithms, *J. Theor. Biol.* 234 (2) (2005) 201–212. Elsevier.
- [33] M.W. Tanner, L. Sattenspiel, L. Ntamo, Finding optimal vaccination strategies under parameter uncertainty using stochastic programming, *Math. Biosci.* 215 (2) (2008) 144–151.
- [34] V.M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, G. Pappas, Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks, in: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), IEEE, 2013, pp. 7486–7491.
- [35] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2007, pp. 862–876.
- [36] J.J. Higgins, Introduction to Modern Nonparametric Statistics, Cengage Learning, 2003.