Universidade Federal de Minas Gerais
Departamento de Matemática

Notas de Aula

Otimização Escalar e Vetorial

Volume 1: Conceitos Preliminares

Professor: Ricardo H. C. Takahashi

Belo Horizonte, Janeiro de 2007

Conteúdo

Parte I

Introdução e Conceitos Preliminares

Capítulo 1

Introdução

Este conjunto de notas de aula tem o objetivo de apresentar o lugar da *otimização multiobjetivo*, ou *otimização vetorial* dentro do problema de *projeto assistido por computador*. Com o objetivo de sistematizar um conjunto de conceitos necessários à apresentação da teoria da otimização vetorial, e ainda para tornar este texto auto-contido, são apresentados alguns capítulos iniciais com a teoria da *otimização escalar* (ou mono-objetivo).

Neste capítulo, discute-se:

1. o que é o problema de *projeto assistido por computador* (PAC) de sistemas;

2. qual é o papel dos mecanismos de otimização dentro dos sistemas de PAC;

3. qual é o papel da abordagem multiobjetivo, no contexto dos problemas de otimização em PAC.

Por fim, é apresentada ainda neste capítulo a formalização do problema de otimização multiobjetivo.

1.1 Otimização em Projeto Assistido por Computador

Os mecanismos de otimização tratam da questão de determinar a "melhor solução" de problemas abstratos para os quais é possível quantificar o grau de adequação de cada solução à necessidade em causa. Assim, esses mecanismos se encontram presentes em uma grande variedade de aparatos técnicos. Alguns contextos nos quais são encontrados mecanismos de otimização são:

- Problemas de gerência (gestão de recursos, gestão da produção, etc). Nesse contexto, a otimização é usualmente denominada “pesquisa operacional”.
- Problemas específicos de projeto de aparatos. Nesse contexto, a teoria de otimização é fundida com alguma outra teoria, dando origem a sub-campos com denominação específica, tais como: teoria de controle ótimo, teoria de identificação de sistemas, etc.
- Sistemas de projeto assistido por computador.

No primeiro caso acima, existe em geral um modelo relativamente grosseiro do processo sob análise. O próprio processo, por sua vez, pode ser bastante complexo, possuindo muitas vezes um grande número de variáveis. Pode ser necessário que a otimização seja executada rapidamente, para utilização dos resultados “em tempo real”. O resultado do mecanismo de otimização é utilizado, em geral, como uma “linha de conduta”, que serve para guiar a ação gerencial, sem ter o papel entretanto de especificar os detalhes dos procedimentos a serem seguidos. Isso tudo repercute no tipo de sistema de otimização que é empregado nesse contexto, que terá freqüentemente características do tipo: pouca necessidade de soluções precisas e alta necessidade de execução veloz.

No segundo caso, a formulação teórica do problema de otimização é sempre embutida em um mecanismo específico para o sistema em questão. Já no momento de formular o algoritmo, o mecanismo de otimização torna-se indivisível do mecanismo de modelagem do sistema, de forma que os algoritmos resultantes são inteiramente específicos para cada caso. Nessas situações, em geral, o modelo do processo é bastante refinado (embora isso não queira necessariamente dizer que seja preciso, no sentido de equivaler ao sistema físico real que está sendo tratado), sendo entretanto também bastante simples, no sentido em que não prevê uma grande diversidade de variações nem uma grande complexidade estrutural (ou seja, não admite modelos com diversos blocos funcionais de naturezas diferentes interligados). Freqüentemente, há um conhecimento suficiente do modelo em questão para que se possam construir algoritmos que com certeza convergem para as soluções ótimas no sentido analítico. Devido à especificidade do algoritmo empregado, este é freqüentemente muito eficiente computacionalmente.

O terceiro caso acima trata-se de uma situação intermediária, entre o primeiro em que são utilizados algoritmos gerais em problemas com formulação grosseira e problemas em que são utilizados algoritmos especializados em problemas com formulação bastante detalhada. Os mecanismos de *projeto assistido por computador* aparecem onde quer que haja modelos de sistemas a serem projetados. Nesse contexto, nem sempre será interessante desenvolver algoritmos específicos para a otimização do processo em questão, uma vez que: (i) o ganho em eficiência computacional pode não compensar o trabalho de formulação teórica de um algoritmo específico e (ii) a complexidade do problema pode ser excessiva, elevando de maneira substancial tal esforço teórico necessário. São utilizados, portanto, algoritmos de propósito geral. Por outro lado, os modelos são freqüentemente bastante refinados (o que não impede a imprecisão, ou seja, a existência de desvios em relação à realidade física). Sua complexidade muitas vezes não permite que se conheça de antemão todas as propriedades desses modelos, de forma que não é trivial o problema de saber se uma solução é a melhor possível, em um problema de otimização. O número de variáveis a otimizar normalmente não é muito grande, o que reduz o custo computacional do processo de otimização. Por outro lado, em geral não é crítico que se obtenham as soluções em curto prazo, de forma que normalmente é possível realizar cálculos extensivos, que viabilizam métodos refinados de busca de soluções.

Estas notas tratam de problemas de otimização nesse terceiro contexto. Serão particularmente enfatizados aqui problemas de otimização de sistemas com modelos não-lineares contínuos.

1.2 Sistemas de Projeto Assistido por Computador

A figura 1.1 mostra um diagrama genérico de um sistema de projeto assistido por computador (PAC).

Esse sistema visa projetar um determinado "dispositivo", para o qual é disponível um modelo computacional parametrizado. O projeto consiste na escolha dos parâmetros adequados do dispositivo, para que o mesmo tenha um comportamento "adequado".

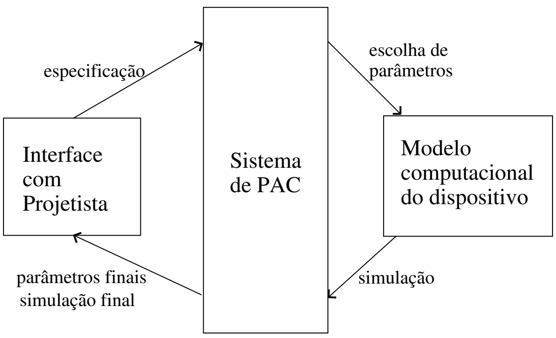

Figura 1.1: Diagrama de blocos de um sistema de projeto assistido por computador genérico.

O sistema físico a ser projetado é modelado através de um conjunto de equações que descrevem as relações entre as variáveis do modelo, e que compõem o bloco "modelo do sistema". O bloco "modelo do sistema" inclui ainda um mecanismo de simulação do funcionamento do sistema, dado um conjunto de parâmetros definidos pelo usuário. O bloco "interface com o usuário" permite ao usuário a definição dos parâmetros do sistema, e devolve ao mesmo a visualização do comportamento do sistema dados aqueles parâmetros (através, por exemplo, de gráficos). Esses dois blocos constituem, juntos, o sistema mais elementar possível de PAC, em que a interação entre o modelo do dispositivo e o projetista é feita diretamente. Este esquema é mostrado na figura 1.2. O projetista neste caso, utilizando sua experiência e

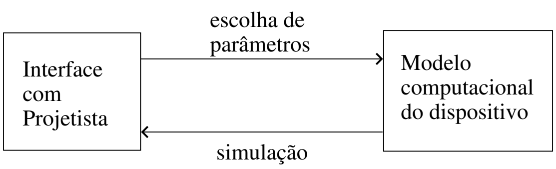

Figura 1.2: Diagrama de blocos de um sistema de projeto assistido por computador primitivo, constituído apenas pelo modelo computacional do dispositivo a ser projetado.

conhecimento, vai variando os valores dos parâmetros do dispositivo até determinar uma combinação considerada adequada. O problema básico desse esquema elementar de PAC é o excesso de opções que se colocam para o projetista. Desde as melhores soluções (que são relativamente poucas) até as muito ruins (que se constituem, geralmente, na maioria das possibilidades de combinações de parâmetros), todas estão indiscriminadamente disponíveis para o projetista. O projetista então não é "guiado" pelo sistema de PAC para a escolha das opções razoáveis, tendo de utilizar apenas seu conhecimento, sua intuição ou sua sorte para procurar uma solução de projeto que seja adequada.

Para munir o projetista de ferramentas que efetivamente o auxiliem na escolha dos parâmetros de projeto adequados, uma primeira abordagem possível é a inclusão de uma base de dados de soluções ou de regras de projeto. Com essa base é possível, por exemplo, partir de soluções já determinadas para problemas anteriormente resolvidos, como solução inicial para o problema em questão. Se se tratar de uma base de regras de projeto, o conhecimento anteriormente adquirido pode ser sistematizado, por exemplo através de mecanismos de inteligência artificial, para auxiliar na triagem das soluções.

Uma outra abordagem seria a definição, em termos quantitativos, do que seriam os "objetivos de projeto". Havendo tal definição quantitativa, seria possível comparar de maneira inequívoca duas possíveis soluções, sendo possível determinar de maneira não-ambígua qual delas é "melhor". Com tal maneira de ordenar as soluções em uma escala de "piores" e "melhores", torna-se possível fazer a própria máquina começar a gerar automaticamente soluções e "caminhar" automaticamente para soluções cada vez melhores. Esta é a abordagem da *otimização*, que será discutida neste trabalho. Neste texto, será estudado especificamente o caso em que o *sistema PAC* é constituído por um *mecanismo de otimização*, como mostrado na figura 1.3.

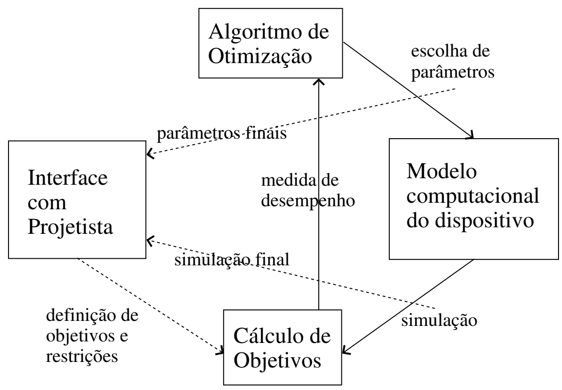

Figura 1.3: Diagrama de blocos de um sistema de projeto assistido por computador à base de um mecanismo de otimização.

1.3 Otimização em PAC

Os sistemas de PAC baseados em otimização são definidos em termos de dois blocos funcionais básicos:

- O modelo do sistema, incluindo agora uma avaliação de uma figura de mérito associada a uma dada implementação do sistema. Esse bloco possui como entradas os parâmetros de projeto do sistema, e como saída aquela figura de mérito.

- Um mecanismo "otimizador", que possui como entrada a figura de mérito obtida do bloco anterior, e que produz como saída um conjunto de parâmetros de projeto (uma tentativa de solução).

Esses blocos funcionais devem interagir autonomamente, sem intervenção do usuário, produzindo ao final do processo uma solução "ótima", ou seja, um conjunto de parâmetros de projeto que minimiza (ou maximiza) a figura de mérito.

1.3.1 A Abordagem Escalar

Classicamente, os sistemas de PAC baseados em mecanismos de otimização têm empregado a abordagem *mono-objetivo*, ou *escalar*. Isso significa que a figura de mérito que o mecanismo de otimização deve minimizar é um funcional (uma função cuja imagem é escalar). Seja $x \in \mathbb{R}^n$ o vetor de parâmetros de projetos que devem ser escolhidos, e $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$ o funcional-objetivo que quantifica a adequação de cada solução x (considere-se aqui a convenção de que quanto menor $f(x)$ melhor será a solução x). O problema de otimização pode ser expresso como:

$$x^* = \arg \min_x f(x) \tag{1.1}$$

ou seja, o mecanismo de otimização deve ser capaz de determinar o vetor x^* que minimiza o funcional $f(x)$. Esta é a formulação característica dos problemas de *otimização irrestrita*.

Na prática, além de minimizar determinada função-objetivo, é necessário atender a certas restrições que correspondem a limitações de natureza física ou tecnológica, ou mesmo ao fato de certas soluções não serem aceitáveis por motivos mais subjetivos. Para expressar tais restrições, definem-se regiões no espaço de parâmetros $\mathbb{R}^n$ através de desigualdades ou de igualdades:

$$\begin{aligned} g_i(x) &\leq 0 \ \ \forall \ i = 1, \ldots, r \\ h_i(x) &= 0 \ \ \forall \ i = 1, \ldots, p \end{aligned} \tag{1.2}$$

As soluções do problema de otimização serão procuradas apenas entre aquelas que atenderem a tais igualdades e desigualdades. O problema de determinação de soluções que atendam às restrições (1.2) é o chamado *problema de factibilidade*.

Um problema de otimização combinado com um problema de factibilidade é a situação mais freqüentemente encontrada na prática, sendo expresso por:

$$x^* = \ \arg\ \min_x \ f(x) \text{sujeito a: } \begin{cases} g_i(x) \leq 0 \ \ \forall \ i = 1, \ldots, r \\ h_i(x) = 0 \ \ \forall \ i = 1, \ldots, p \end{cases} \tag{1.3}$$

A formulação (1.3) corresponde à chamada *otimização restrita*.

Sabe-se que os sistemas com significado prático devem ser projetados para atender a múltiplos critérios de projeto, não necessariamente comparáveis entre si. Considere-se um problema de projeto com m objetivos distintos:

$$\begin{aligned} x_i^* = \ \arg\ \min_x \ f_i(x) \ \ i = 1, \ldots, m \\ \text{sujeito a: } \begin{cases} g_i(x) \leq 0 \ \ \forall \ i = 1, \ldots, r \\ h_i(x) = 0 \ \ \forall \ i = 1, \ldots, p \end{cases} \end{aligned} \tag{1.4}$$

Esse problema conduzirá a m soluções distintas, cada qual atendendo da melhor forma possível a um dos objetivos. A questão que se coloca para o projetista é: será possível escolher apenas um dos objetivos de projeto para implementar, ignorando os demais objetivos, ou haverá uma maneira de considerar simultaneamente todos os objetivos distintos?

EXEMPLO 1.1 *Determinado equipamento deve ser fabricado a um custo mínimo de produção, mas deve também apresentar o melhor desempenho possível. Evidentemente, o projeto mais barato possível tende a não apresentar o melhor desempenho, assim como o projeto que visa ao melhor desempenho possível tende a não ser o mais barato.* ♢

Posta essa dificuldade de lidar com objetivos que em geral são conflitantes, dentro da abordagem clássica de otimização há um expediente que é em geral utilizado para tratar de múltiplos objetivos: faz-se uma soma ponderada dos objetivos, atribuindo maior ponderação àqueles considerados de maior importância, e menor ponderação àqueles considerados de menor importância. Com isso, constitui-se uma nova função objetivo que agora é um escalar, composto de parcelas que correspondem aos múltiplos objetivos:

$$x^* = \arg\ \min_x \sum_{i=1}^{m} \lambda_i\ f_i(x)$$
$$\text{sujeito a: } \begin{cases} g_i(x) \leq 0 \ \ \forall\ i = 1, \ldots, r \\ \\ h_i(x) = 0 \ \ \forall\ i = 1, \ldots, p \end{cases} \tag{1.5}$$

Este problema possui uma solução única, que foi gerada a partir de considerações que incluíram todos os diferentes objetivos. Em princípio, variando-se o valor dos diversos λ_i, são obtidas diferentes soluções que ponderam diferentemente os objetivos individuais. O problema é solucionável por qualquer método de otimização mono-objetivo.

Esse método de lidar com múltiplos objetivos funciona, apresentando resultados satisfatórios, em muitos contextos. Em diversos outros contextos, entretanto, pode-se perceber que o método falha, se seus resultados forem submetidos a uma análise mais sutil. O ferramental para realizar tal análise, assim como para produzir as soluções adequadas, é fornecido pela *otimização multiobjetivo*, ou *otimização vetorial*.

EXEMPLO 1.2 *Para mostrar o tipo de dificuldade que é possível haver na utilização da abordagem mono-objetivo para a otimização de múltiplos objetivos, é aqui mostrado um exemplo em que tal abordagem falha completamente. Este exemplo será comentado em maior detalhe no momento adequado. Considerem-se as duas funções de uma única variável:*

$$f_1(x) = 1 - \exp\left(\frac{-(x-4)^2}{9}\right)$$

$$f_2(x) = 1 - \exp\left(\frac{-(x+4)^2}{9}\right)$$

Suponha-se, para fins de raciocínio, que tratam-se de duas funções significando o custo de uma peça (f_1) e o desempenho da peça (f_2), de acordo com uma única variável x considerada como parâmetro de projeto da peça.

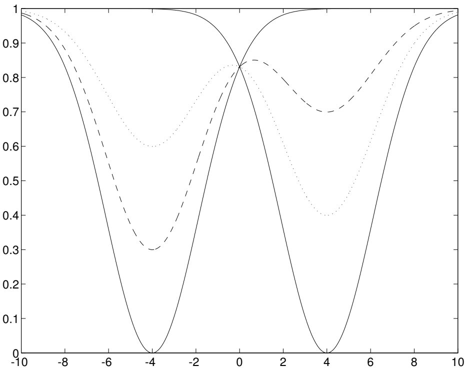

Figura 1.4: Representação das funções-objetivo f_1, f_2 e das funções ponderadas $0.7f_1 + 0.3f_2$ e $0.4f_1 + 0.6f_2$.

Ambas as funções estão representadas no gráfico da figura 1.4 em traço contínuo. A função dada por:

$$F_1(x) = 0.7f_1(x) + 0.3f_2(x)$$

aparece representada no mesmo gráfico em traço-traço. Se esta for a função-objetivo escolhida, isto significa que o custo tem maior peso, e o desempenho menor importância para o projetista. A função:

$$F_2(x) = 0.4f_1(x) + 0.6f_2(x)$$

aparece em ponto-ponto. A escolha de tal função-objetivo significaria que o projetista atribui maior importância ao desempenho, e uma importância um pouco menor ao custo.

Pode-se perceber que $F_1(x)$ *possui mínimo que coincide com o de* $f_1(x)$*, e que* $F_2(x)$ *possui mínimo que coincide com o de* $f_2(x)$*. De fato, a solução de qualquer problema formulado como:*

$$\min_x \ \alpha f_1(x) + (1-\alpha)f_2(x)$$

irá sempre produzir como resultados apenas os pontos $x = 4$ *e* $x = -4$*, ou seja, os mínimos individuais de* $f_1(x)$ *ou de* $f_2(x)$*. Se o projetista se basear nos resultados desse esquema de otimização, suas únicas alternativas serão: ou escolher a solução de mínimo custo (mínimo* f_1*) ou então escolher a solução de máximo desempenho (mínimo* f_2*). O projetista não terá à sua disposição nenhuma alternativa intermediária.* $\diamondsuit$

1.3.2 A Abordagem Vetorial

A abordagem de *otimização multiobjetivo*, ou *otimização vetorial*, parte da constatação de que, havendo diversos objetivos em questão num problema de projeto, haverá soluções de dois tipos:

1. Haverá soluções que, sob todos os objetivos simultaneamente considerados, serão suplantadas por outras soluções. Isso significa que há soluções que conseguem valores menores de função objetivo para todos os m objetivos $f_i(\cdot)$. Essas soluções, naturalmente, devem ser descartadas, pois há soluções melhores.

2. Haverá ainda soluções que, comparadas com outras soluções, serão melhores em algum ou alguns objetivos, mas piores em outro ou outros objetivos. Isso significa que o projetista, quando se deparar com um par dessas soluções, deverá considerar a hipótese de adotar uma ou outra (não deverá haver o descarte automático de nenhuma das duas), sendo que a escolha provavelmente será realizada com algum grau de subjetividade.

As soluções desse último grupo são denominadas *soluções eficientes* ou *soluções Pareto-ótimas*. O conjunto dessas soluções é um objeto bem determinado, posto um problema de otimização com múltiplos objetivos. A determinação, no todo ou em parte, desse conjunto, é um dos problemas centrais da otimização vetorial.

Deve-se observar que esse problema faz sentido exatamente porque em princípio tende a não existir uma solução única x^* que simultaneamente minimiza todas as diferentes funções-objetivo. Um problema com mais de um objetivo tende a possuir um conjunto, em geral limitado, com infinitas soluções.

Para o objetivo de resolver um problema de projeto de determinado sistema, deve-se convergir para uma única solução final. O conjunto Pareto-ótimo constitui um elenco de alternativas, todas candidatas a se tornarem essa solução final. Para viabilizar a síntese final do sistema pelo projetista, a otimização multiobjetivo emprega ainda uma outra etapa, na qual o conjunto de soluções-candidatas é reduzido até a determinação de uma única solução, através de uma sistemática de interação com o projetista (ou decisor). Essa etapa, que inclui uma sistemática de busca (em uma trajetória sobre o conjunto Pareto-ótimo), supõe a existência dentro do decisor de uma chamada *função utilidade*, que significa um padrão de preferências coerente e ordenado.

O problema multiobjetivo (PMO) pode então ser formulado como uma combinação dessas etapas de determinação das soluções eficientes e de escolha da solução final (aplicação da função utilidade).

1.4 Formulação do Problema de Otimização Vetorial

Nesta seção, é apresentada preliminarmente a formulação completa do problema de otimização vetorial, incluindo as etapas de determinação do conjunto de soluções Pareto-ótimas e de aplicação a esse conjunto da função utilidade. Essas formulações serão exploradas em detalhe nos capítulos subseqüentes.

1.4.1 Etapa de Determinação das Soluções Eficientes

Seja $x \in \mathbb{R}^n$ o vetor dos parâmetros de projeto de um problema multiobjetivo (PMO), e seja $\mathcal{F}_x \subset \mathbb{R}^n$ um subconjunto do espaço ao qual o vetor x se encontra restrito. Seja ainda $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$ o vetor de funções-objetivo desse problema. Dados $f(\cdot)$ e $\mathcal{F}_x$, o conjunto $\mathcal{X}^*$ das *soluções eficientes* do PMO descrito por:

$$\begin{aligned} &\min_x \quad f(x) \\ &\text{sujeito a: } \left\{ x \in \mathcal{F}_x \right. \end{aligned} \tag{1.6}$$

fica definido univocamente. Naturalmente, é necessário definir o que significa minimizar um vetor de funções, o que será feito no capítulo 10. Uma caracterização operacional do conjunto $\mathcal{X}^*$ é dada por:

$$\mathcal{X}^* \triangleq \{x \in \mathcal{F}_x \mid \;\nexists x_1 \in \mathcal{F}_x \text{ tal que } f(x_1) \leq f(x) \text{ e } f(x_1) \neq f(x)\} \tag{1.7}$$

Para a expressão (1.7) fazer sentido, é necessário definir os operadores relacionais $\leq$ e $<$ envolvendo vetores, o que também será feito no capítulo 10.

Algoritmos de determinação das soluções eficientes de um PMO serão formulados, a partir das propriedades do conjunto $\mathcal{X}^*$ que estão implícitas na definição (1.7).

1.4.2 Etapa de Decisão

Uma vez que se tenha determinado o conjunto das soluções eficientes do problema, passa-se à etapa de escolha de uma solução final dentre essas opções[1].

O *decisor* real, que efetivamente irá escolher a alternativa final de projeto, pode ser um ser humano "projetista", ou toda uma equipe de projeto. As alternativas de projeto devem ser apresentadas ao decisor, para que o mesmo selecione aquela que efetivamente será implementada.

Caso o conjunto de soluções eficientes do problema envolva um conjunto pequeno de alternativas, este pode ser todo apresentado ao decisor, para que seja tomada uma decisão com base na avaliação simultânea de todo o universo de possibilidades "eficientes". No entanto, este não é usualmente o caso. Freqüentemente, existe uma quantidade muito grande de alternativas, que não podem ser processadas simultaneamente pelo ser humano. É necessária, portanto, uma sistemática de apresentação das alternativas, segundo uma seqüência que permita o afunilamento das escolhas de maneira consistente, garantindo ao mesmo tempo que:

- O número de consultas ao decisor será mantido no menor patamar possível, sendo que as respostas a cada consulta serão utilizadas para eliminar uma grande quantidade de alternativas;

- A cada consulta ao decisor, serão apresentados dados em quantidade tal que um ser humano consiga lidar simultaneamente, sendo evitada a apresentação de dados em excesso que impossibilitem a formulação de uma decisão racional;

- As melhores alternativas não serão perdidas, sendo que a sistemática de decisão permitirá, de maneira determinística, encontrar a mesma solução que seria alcançada caso o decisor fosse apresentado a todas as alternativas simultaneamente, e caso estivesse ao seu alcance processar todas essas alternativas simultaneamente.

---

[1] É possível que a interação entre o decisor e o sistema de projeto assistido por computador ocorra de tal forma que sejam concomitantes as etapas de determinação de soluções eficientes e de escolha de decisão. Nesse caso, cada solução eficiente que for determinada será sumetida ao decisor, cuja "opinião" direcionará a escolha de uma nova solução eficiente. Esse *modelo de decisão* possivelmente envolve um menor número de avaliações das funções-objetivo, e um maior número de consultas ao decisor.

O sistema de consultas ao decisor é o objeto da chamada *etapa de decisão* da otimização multiobjetivo.

Capítulo 2

Definições de Referência

Neste capítulo são estabelecidas diversas definições matemáticas, que são utilizadas para construir os conceitos que serão empregados no restante do trabalho. Essas definições não são diretamente relacionadas com o tema da "otimização", sendo antes definições comuns a contextos matemáticos diversos. Por este motivo, foram agrupadas neste capítulo introdutório.

As definições aqui apresentadas são provenientes das fontes citadas. Por vezes, a definição é adaptada, tanto em termos formais quanto de notação, com o objetivo de assegurar a consistência metodológica do conjunto dos conceitos formulados, bem como a coerência com o material desenvolvido no restante deste trabalho. Entretanto, a definição apresentada é sempre equivalente à encontrada na fonte original. Este capítulo apenas compila conceitos já definidos na literatura, tendo sido incluído com o objetivo de tornar preciso o significado dos termos a serem empregados no restante do trabalho.

2.1 Espaços e Normas

As definições contidas nesta seção seguem as referências (Chen 1984) e (Vidyasagar 1993).

Definição 2.1 (Corpo) *Um corpo consiste num conjunto, denotado* $\mathbb{F}$*, de elementos denominados escalares, e duas operações chamadas adição* ($+ : \mathbb{F} \times \mathbb{F} \mapsto \mathbb{F}$) *e multiplicação* ($\cdot : \mathbb{F} \times \mathbb{F} \mapsto \mathbb{F}$)*, satisfazendo as seguintes condições* $\forall\ \alpha, \beta, \gamma \in \mathbb{F}$*:*

(C1) $\alpha + \beta \in \mathbb{F}$ *e* $\alpha \cdot \beta \in \mathbb{F}$.

(C2) $\alpha + \beta = \beta + \alpha$ *e* $\alpha \cdot \beta = \beta \cdot \alpha$.

(C3) $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$ *e* $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$.

(C4) $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$.

(C5) $\mathbb{F}$ *contém um elemento denotado por* 0 *e um elemento denotado por* 1 *tais que:* $\alpha + 0 = \alpha$ *e* $\alpha \cdot 1 = \alpha$.

(C6) *A todo* $\alpha \in \mathbb{F}$ *corresponde um elemento* $\beta \in \mathbb{F}$ *tal que* $\alpha + \beta = 0$.

(C7) *A todo* $\alpha \neq 0 \in \mathbb{F}$ *corresponde um elemento* $\beta \in \mathbb{F}$ *tal que* $\alpha \cdot \beta = 1$.

$\square$

Definição 2.2 (Espaços Vetoriais Lineares) *Um espaço vetorial linear é caracterizado por um conjunto* $\mathcal{V}$*, um corpo* $\mathbb{F}$ *e duas operações, a operação adição vetorial* $(+ : \mathcal{V} \times \mathcal{V} \mapsto \mathcal{V})$ *e a operação multiplicação por escalar* $(\cdot : \mathbb{F} \times \mathcal{V} \mapsto \mathcal{V})$*, tais que os seguintes axiomas sejam válidos* $\forall\ x, y, z \in \mathcal{V}$ *e* $\forall\ c, c_1, c_2 \in \mathbb{F}$*:*

(V1) $x + y = y + x$.

(V2) $x + (y + z) = (x + y) + z$.

(V3) *Existe um elemento* $\mathbf{0} \in \mathcal{V}$ *tal que* $x + \mathbf{0} = \mathbf{0} + x = x, \quad \forall x \in \mathcal{V}$.

(V4) *Para cada* $x \in \mathcal{V}$ *existe um elemento denotado por* $-x \in \mathcal{V}$ *tal que* $x + (-x) = \mathbf{0}$.

(V5) $c_1 \cdot (c_2 \cdot x) = (c_1 \cdot c_2) \cdot x$.

(V6) $c \cdot (x + y) = c \cdot x + c \cdot y$.

(V7) $(c_1 + c_2) \cdot x = c_1 \cdot x + c_2 \cdot x$.

(V8) $1 \cdot x = x$.

$\square$

Definição 2.3 (Subespaço Vetorial) *Denomina-se* subespaço vetorial *a qualquer subconjunto $\mathcal{M}$ de um espaço vetorial linear $\mathcal{V}$ (sobre o corpo $\mathbb{F}$) tal que:*

1. $x, y \in \mathcal{M} \Rightarrow x + y \in \mathcal{M}$
2. $x \in \mathcal{M},\ c \in \mathbb{F} \Rightarrow c \cdot x \in \mathcal{M}$.

$\square$

Definição 2.4 (Norma) *Seja $\mathcal{V}$ um espaço vetorial sobre o corpo dos reais. Uma função $F : \mathcal{V} \mapsto \mathbb{R}$ com imagem no conjunto dos números reais e domínio no conjunto $\mathcal{V}$ é denominada uma* norma *dos elementos de $\mathcal{V}$ se satisfaz as seguintes propriedades $\forall\ x, y \in \mathcal{V}$ e $\forall\ \alpha \in \mathbb{R}$:*

(N1) $F(x) \geq 0, F(x) = 0 \Leftrightarrow x = \mathbf{0}$.

(N2) $F(\alpha x) = |\alpha|\, F(x)$.

(N3) $F(x + y) \leq F(x) + F(y)$.

A notação $\|x\|$ será empregada genericamente neste texto para indicar $F(x)$. O contexto deverá deixar claro a que espaço e a que norma específicos tal notação se refere.

$\square$

Definição 2.5 (Normas Vetoriais p) *Seja $x \in \mathbb{R}^n$ um vetor real de dimensão finita, e seja x_i sua i-ésima componente escalar. Sua norma p, denotada por $\|x\|_p$, é definida pela expressão:*

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \quad , \ 1 \leq p < \infty \tag{2.1}$$

Para $p = \infty$ a definição é:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \tag{2.2}$$

$\square$

EXEMPLO 2.1 *A norma euclideana de um vetor é um exemplo de uma norma p, para $p = 2$:*

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{2.3}$$

$\diamondsuit$

Definição 2.6 (Q-norma Vetorial) *Seja $x \in \mathbb{R}^n$ um vetor de dimensão finita, e seja $Q \in \mathbb{R}^{n\times n}$ uma matriz simétrica definida positiva, ou seja, $Q = Q' > 0$. A* Q-norma *de x é definida como $\|x'Qx\|_2$, ou seja, a norma vetorial 2 do vetor $x'Qx$.* $\square$

Definição 2.7 (Espaços Normados) *Um* espaço normado *$\mathcal{E}$ é um par ordenado $(\mathcal{V}, F)$ em que $\mathcal{V}$ é um espaço vetorial sobre um corpo $\mathbb{F}$ e $F : \mathcal{V} \mapsto \mathbb{R}$ é uma norma aplicável aos elementos de $\mathcal{V}$. O espaço normado $\mathcal{E}$ é definido como:*

$$x \in \mathcal{E} \Leftrightarrow (F(x) < \infty, x \in \mathcal{V}) \tag{2.4}$$

$\square$

Definição 2.8 (Convergência de Seqüência) *Seja uma seqüência ordenada de vetores $\{x_i,\ i = 1, \ldots, \infty\}$ pertencentes a um espaço vetorial normado $(\mathcal{X}, \|\cdot\|)$. Diz-se que essa seqüência* converge *para $x_0 \in \mathcal{X}$ se, para todo escalar $\epsilon > 0$, existe um inteiro $N = N(\epsilon)$ tal que:*

$$\|x_i - x_0\| < \epsilon\ , \quad \forall\ i \geq N \tag{2.5}$$

$\square$

Definição 2.9 (Seqüência de Cauchy) *Seja uma série ordenada de vetores $\{x_i,\ i = 1, \ldots, \infty\}$ pertencentes a um espaço vetorial normado $(\mathcal{X}, \|\cdot\|)$. Essa série é denominada uma* seqüência de Cauchy *se, para todo escalar $\epsilon > 0$, existe um inteiro $N = N(\epsilon)$ tal que:*

$$\|x_i - x_j\| < \epsilon \quad \forall\ i,\ j > N \tag{2.6}$$

$\square$

Definição 2.10 (Espaço de Banach) *Um espaço linear normado $(\mathcal{X}, \|\cdot\|)$ é dito ser* completo*, ou um* espaço de Banach*, se toda seqüência de Cauchy em $(\mathcal{X}, \|\cdot\|)$ converge para um elemento de $\mathcal{X}$.*

$\square$

EXEMPLO 2.2 *Um exemplo de* espaço de Banach *é o espaço* $\mathbb{R}^n$*, no qual toda seqüência de Cauchy converge para um ponto desse mesmo espaço. Um exemplo de espaço que* não é espaço de Banach *é o espaço* $\mathbb{Q}^n$*, dos vetores sobre o corpo dos números racionais. Claramente, algumas seqüências em* $\mathbb{Q}$*, que corresponde ao caso* $n = 1$*, convergem para números irracionais, que se encontram fora de* $\mathbb{Q}$*.*

$\diamondsuit$

2.2 Espaços Topológicos

As definições a seguir são extraídas principalmente de (Rudin 1991).

Um conjunto S é um subconjunto de um conjunto C se:

$$x \in S \Rightarrow x \in C \tag{2.7}$$

o que se representa por $S \subset C$ ou $C \supset S$. De especial interesse neste curso são conjuntos (subconjuntos) do espaço $\mathbb{R}^n$ de vetores reais de dimensão n.

Definição 2.11 (Espaços Topológicos) *Um espaço topológico é um conjunto* $\mathcal{S}$ *para o qual uma coleção* τ *de subconjuntos, denominados* conjuntos abertos *foi especificada, com as seguintes propriedades:*

i. $\mathcal{S}$ *é aberto;*

ii. $\emptyset$ *é aberto;*

iii. a interseção de dois conjuntos abertos quaisquer é um conjunto aberto;

iv. a união de dois conjuntos abertos quaisquer é um conjunto aberto.

$\square$

Uma possível maneira de definir conjuntos abertos em espaços vetoriais normados, para estabelecer topologias nos mesmos, é apresentada a seguir:

Definição 2.12 (Conjunto Aberto em Espaços Normados) *Seja* $\mathcal{X}$ *um conjunto contido em um espaço normado. O conjunto* $\mathcal{X}$ *é dito ser* aberto *se:*

$$x_o \in \mathcal{X} \Rightarrow \exists\ \epsilon \mid x \in \mathcal{X}\ \forall\ \|x - x_o\| < \epsilon \tag{2.8}$$

$\square$

Definição 2.13 (Vizinhança) *Seja um ponto x contido em um espaço topológico. Uma vizinhança de x é qualquer conjunto aberto que contenha x.*

□

Definição 2.14 (Complemento de Conjunto) *Seja $\mathcal{X} \subset \mathcal{Q}$ um conjunto contido em outro conjunto $\mathcal{Q}$. O conjunto $\mathcal{Y}$ é o* complemento *de $\mathcal{X}$ em relação a $\mathcal{Q}$ se:*

$$\begin{aligned} \mathcal{X} \cup \mathcal{Y} &= \mathcal{Q} \\ \mathcal{X} \cap \mathcal{Y} &= \emptyset \end{aligned} \tag{2.9}$$

□

Definição 2.15 (Conjunto Fechado) *Seja $\mathcal{X}$ um conjunto contido em um espaço topológico. O conjunto $\mathcal{X}$ é dito ser* fechado *se seu complemento em relação ao espaço for aberto.*

□

Definição 2.16 (Conjunto Compacto) *Seja o conjunto $Q \subset \mathbb{R}^n$. Esse conjunto é dito* compacto *se for fechado e se dados quaisquer dois pontos $x_1, x_2 \in Q$, eles se encontram a uma distância finita um do outro:*

$$\|x_1 - x_2\| = \delta < \infty \tag{2.10}$$

□

Definição 2.17 (Fecho de Conjunto) *Seja $\mathcal{X}$ um conjunto contido em um espaço topológico. O* fecho *de $\mathcal{X}$, denotado por $\overline{\mathcal{X}}$, é definido como a interseção de todos os conjuntos fechados que contêm $\mathcal{X}$.*

□

Definição 2.18 (Interior de Conjunto) *Seja $\mathcal{X}$ um conjunto contido em um espaço topológico. O* interior *de $\mathcal{X}$, denotado por $\mathcal{X}^\circ$, é definido como a união de todos os conjuntos abertos contidos em $\mathcal{X}$.*

□

Definição 2.19 (Fronteira de Conjunto) *Seja $\mathcal{X}$ um conjunto contido em um espaço topológico. A* fronteira *de $\mathcal{X}$, denotada por $\partial\mathcal{X}$, é definida como a diferença entre o fecho de $\mathcal{X}$ e seu interior:*

$$\begin{aligned} \partial\mathcal{X} &\subset \overline{\mathcal{X}} \\ \partial\mathcal{X} \cap \mathcal{X}^\circ &= \emptyset \end{aligned} \tag{2.11}$$

□

Definição 2.20 (Conjunto Conexo) *Seja o conjunto $Q \subset \mathbb{R}^n$. Esse conjunto é dito* conexo *se, dados quaisquer dois pontos $x_1, x_2 \in Q$, existe uma curva contínua finita q tal que $x_1 \in q$, $x_2 \in q$, e $q \subset Q$.* □

Definição 2.21 (Conjunto Convexo) *Seja um conjunto $\mathcal{X}$, contido em um espaço vetorial, e seja a função $x(x_1, x_2, \alpha)$ definida por:*

$$x(x_1, x_2, \alpha) = \alpha x_1 + (1 - \alpha) x_2 \tag{2.12}$$

para x_1 e x_2 elementos do espaço vetorial e α um escalar. Se se verificar:

$$\{x_1, x_2 \in \mathcal{X} \ ; \ 0 \leq \alpha \leq 1\} \Rightarrow x(x_1, x_2, \alpha) \in \mathcal{X} \tag{2.13}$$

então o conjunto $\mathcal{X}$ é dito ser convexo.

□

Proposição 2.1 (Propriedades de Conjuntos Convexos) *Dados dois conjuntos convexos A e B, então são também conjuntos convexos:*

i. αA, $\forall \ \alpha \in \mathbb{R}$

ii. $A \cap B$

iii. $A + B \triangleq \{x : x = a + b \ , \ a \in A, b \in B\}$

□

O conjunto vazio $\{\emptyset\}$ é definido como convexo.

Definição 2.22 (Fecho Convexo) *O fecho convexo de um conjunto $\mathcal{X}$, denotado por conv $\overline{\mathcal{X}}$, é definido como a interseção de todos os conjuntos convexos que contêm o conjunto $\overline{\mathcal{X}}$.*

□

Definição 2.23 (Casca Convexa) *A* casca convexa *$\mathcal{C}(S)$ de um conjunto S é definida como o menor conjunto convexo que contém S, ou seja:*

i. $\mathcal{C}(S)$ *é um conjunto convexo*

ii. $x \in S \Rightarrow x \in \mathcal{C}(S)$

iii. $\nexists D$ *convexo tal que:*

$$D \supset S$$

$$D \subset \mathcal{C}(S)$$

$$D \neq \mathcal{C}(S)$$

$\square$

2.3 Cones

Um conjunto de grande interesse é o cone.

Definição 2.24 (Cone) *Um conjunto C é um cone se:*

$$x \in C \Rightarrow (\alpha x) \in C \quad \forall\ \alpha \geq 0 \tag{2.14}$$

$\square$

Definição 2.25 (Cone Gerado) *O* cone gerado *pelos vetores x e y é o conjunto definido por:*

$$C = \{z : z = \alpha x + \beta y,\ \forall\ \alpha, \beta \geq 0\} \tag{2.15}$$

$\square$

Definição 2.26 (Cone Polar) *O* cone polar *associado a um conjunto C, denotado por C^*, é o conjunto definido por:*

$$C^* = \{y : y'x \geq 0\ ,\ \forall\ x \in C\} \tag{2.16}$$

$\square$

2.4 Hiperplanos e Poliedros

Um conjunto convexo particularmente importante é o hiperplano.

Definição 2.27 (Hiperplano) *Seja c um vetor diferente de 0 e α um escalar qualquer. Então o conjunto:*

$$H = \{x : c'x = \alpha\} \tag{2.17}$$

define um hiperplano. □

A cada hiperplano correspondem semi-espaços abertos e fechados:

$$\begin{aligned}
&\text{semi-espaços fechados} \begin{cases} H_{\geq} \triangleq \{x : c'x \geq \alpha\} \\ \\ H_{\leq} \triangleq \{x : c'x \leq \alpha\} \end{cases} \\
&\text{semi-espaços abertos} \begin{cases} H_{>} \triangleq \{x : c'x > \alpha\} \\ \\ H_{<} \triangleq \{x : c'x < \alpha\} \end{cases}
\end{aligned} \tag{2.18}$$

Note que para qualquer hiperplano H:

$$\mathbb{R}^n = H_{\leq} \cup H_{>} = H_{\geq} \cup H_{<} \tag{2.19}$$

Definição 2.28 (Politopo e Poliedro) *A interseção de um número finito de semi-espaços fechados é chamada de politopo. Um politopo limitado é chamado de poliedro.* □

Definição 2.29 (Hiperplano Suporte) *Um hiperplano H que contém pelo menos um ponto da fronteira de um conjunto C e que situa esse conjunto C inteiramente no interior de um de seus semi-espaços fechados é chamado de* hiperplano suporte *de C.* □

Em outras palavras, se C admite um hiperplano suporte em um ponto x^0 de sua fronteira, então existem um vetor $c \neq 0$ e um escalar $\alpha \in \mathbb{R}$ tais que:

$$H = \left\{x : c'x^0 = \alpha\right\} \ \ e \ \ \{C \subset H_{\leq} \ \ ou \ \ C \subset H_{\geq}\} \tag{2.20}$$

Conjuntos convexos fechados podem ser caracterizados pela interseção de todos os semi-espaços fechados que os contêm.

Capítulo 3

Caracterização das Funções

O problema de otimização, na sua formulação mono-objetivo, pode ser definido como:

$$
x^* = \arg \min_x f(x)
$$
$$
\text{sujeito a:} \begin{cases} g_i(x) \leq 0 \; ; \quad i = 1, \ldots, r \\ \\ h_j(x) = 0 \; ; \quad j = 1, \ldots, p \end{cases} \tag{3.1}
$$

sendo que $x \in \mathbb{R}^n$, $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^1$, $g(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^r$, e $h(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^p$.

A escolha de técnicas adequadas para tratar esse problema depende da natureza das funções $f(x)$, $g(x)$ e $h(x)$. Não há uma técnica de otimização que seja "universal", no sentido de ser a melhor técnica para otimizar quaisquer funções. O objetivo deste capítulo é o de formular caracterizações para essas funções que sejam relevantes para a construção e análise de algoritmos para sua otimização.

Neste capítulo são postas as seguintes questões, relacionadas com a questão de *o quê são* as soluções do problema (3.1):

1. Dado o funcional[1] $f(\cdot)$, o que são os pontos de mínimo desse funcional, ou seja, o que são as soluções do problema de otimização?

2. O que são os pontos de mínimo desse funcional, se são dadas também as restrições $g_i(x) \leq 0$ e $h_i(x) = 0$?

[1] Um *funcional* é uma função de um espaço vetorial em seu próprio corpo, ou seja, que retorna um único valor (um número escalar) cuja natureza é a mesma das componentes do argumento. Ou seja, se $f(\cdot)$ é um funcional real então $f : \mathbb{R}^n \mapsto \mathbb{R}^1$.

3. Dado um ponto $x \in \mathbb{R}^n$, que tipo de testes podem ser realizados para determinar se esse ponto é ou não um ponto de mínimo de $f(\cdot)$, nos dois casos acima?

Respostas a essas questões serão fornecidas tanto num sentido local (mínimos locais) quanto global (mínimos globais).

Além de caracterizar as soluções do problema (3.1), é necessário agregar alguma informação que seja útil para se decidir *como proceder* para encontrar tais soluções. Algumas caracterizações úteis, definidas a seguir neste capítulo, classificam as funções em:

1. Unimodais ou multimodais;
2. Contínuas ou descontínuas;
3. Diferenciáveis ou não diferenciáveis;
4. Convexas, quasi-convexas ou não convexas.

Cada uma dessas informações a respeito da função, se estiver disponível, permite a agregação de um certo tipo de informação de caráter global que auxilia o processo de otimização.

3.1 Superfícies de Nível e Modalidade

A caracterização de funções adotada aqui se fundamenta nos conceitos de *superfície de nível* e de *região sub-nível*.

Definição 3.1 (Superfície de Nível) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. A* superfície de nível *$S(f, \alpha)$, associada ao nível α, é definida como:*

$$S(f, \alpha) = \{x \in C \mid f(x) = \alpha\} \tag{3.2}$$

$\square$

Definição 3.2 (Região Sub-Nível) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. A* região sub-nível *$R(f, \alpha)$, associada ao nível α, é definida como:*

$$R(f, \alpha) = \{x \in C \mid f(x) \leq \alpha\} \tag{3.3}$$

$\square$

Normalmente $S(f,\alpha)$ corresponde a uma fronteira de $R(f,\alpha)$, embora seja possível escolher C de forma que isso não ocorra. Claramente é válida uma relação de ordenação das regiões de sub-nível de uma função:

Proposição 3.1 *Seja* $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. *As regiões de sub-nível dessa função obedecem a:*

$$R(f,\alpha_1) \supset R(f,\alpha_2) \;\Leftrightarrow\; \alpha_1 > \alpha_2 \tag{3.4}$$

$\square$

Podem-se pensar os problemas de otimização como sendo equivalentes a um problema de determinar pontos que estejam sucessivamente no interior de regiões de sub-nível cada vez inferiores (de menor valor de α). Em linhas gerais, se se constroem algoritmos que produzem tais seqüências de pontos, produz-se uma "contração" do conjunto definido pelas regiões de sub-nível, sendo a solução atingida quando a região de sub-nível se degenerar nos pontos de ótimo.

As regiões de sub-nível, analisadas sob o ponto de vista topológico, definem uma categorização importante para as funções:

Definição 3.3 (Função Unimodal) *Seja* $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. *Diz-se que* $f(\cdot)$ *é* unimodal *se* $R(f,\alpha)$ *é conexo para todo* $\alpha \in \mathbb{R}$. *Diz-se ainda que* $f(\cdot)$ *é* estritamente unimodal *se, além disso,* $R(f,\alpha)$ *é um conjunto compacto para todo* $\alpha \in \mathbb{R}$. $\square$

Por simetria, define-se ainda:

Definição 3.4 (Função Multimodal) *Seja* $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. *Diz-se que* $f(\cdot)$ *é* multimodal *se existe* $\alpha \in \mathbb{R}$ *tal que* $R(f,\alpha)$ *não é conexo.* $\square$

NOTA 3.1 *Note-se que uma função unimodal pode possuir múltiplos mínimos, desde que o conjunto destes forme um conjunto conexo, e uma função estritamente unimodal também pode possuir múltiplos mínimos, desde que o conjunto destes forme um conjunto conexo compacto. O primeiro caso ocorre, por exemplo, para a função:*

$$f(x) = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

para a qual todos os pontos que pertencem ao eixo $x_1 = 0$ *(esses pontos formam um conjunto conexo mas não compacto) constituem mínimos. Essa observação revela* *uma diferença fundamental das noções de função unimodal e de função multimodal aqui definidas em relação às definições usualmente encontradas na literatura. O presente autor acredita que, no formato apresentado neste texto, essas definições ganham maior funcionalidade para articularem a teoria de otimização.*

$\diamondsuit$

3.1.1 Bacias de Atração

Ao redor de mínimos locais, sempre haverá regiões nas quais a função se comportará de maneira unimodal. Tais regiões são definidas como *bacias de atração* associadas a tais mínimos. Para estabelecer essa definição, é necessário definir preliminarmente a *região conexa de sub-nível*.

Definição 3.5 (Região Conexa de Sub-Nível) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$, seja a região de sub-nível $R(f, \alpha)$, associada ao nível α, e seja um ponto $x_o \in R(f, \alpha)$. A* região conexa de sub-nível *$R_c(f, \alpha, x_o)$ é definida como o maior subconjunto conexo de $R(f, \alpha)$ que contém x_o.* $\square$

Agora é possível definir *bacia de atração*:

Definição 3.6 (Bacia de Atração) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$, e seja $x^* \in C$ um mínimo local de $f(\cdot)$. A* bacia de atração *de x^* é definida como a maior região conexa de sub-nível associada a x^*, sendo α^* o nível correspondente, tal que a função restrita a essa região:*

$$f(\cdot) : R_c(f, \alpha^*, x^*) \mapsto \mathbb{R} \tag{3.5}$$

é unimodal. A bacia de atração é dita estrita *se nessa região a função é estritamente unimodal.* $\square$

3.2 Continuidade e Diferenciabilidade

Suposições de continuidade e de diferenciabilidade das funções permitem extrair conseqüências interessantes a respeito de suas superfícies de nível e bacias de atração.

Proposição 3.2 *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. Se $f(\cdot)$ é contínua no domínio C, então*

$$dist(S(f, \alpha_1), S(f, \alpha_2)) > 0 \ \forall \ (\alpha_1, \alpha_2) \mid |\alpha_1 - \alpha_2| > 0 \tag{3.6}$$

sendo $dist(\cdot, \cdot)$ a função distância. $\square$

Corolário 3.1 *Superfícies de nível de funções contínuas não se tocam nem se cruzam.* □

Proposição 3.3 *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. Se $f(\cdot)$ é diferenciável no domínio C, então toda superfície de nível $S(f, \alpha)$ é suave, sendo o hiperplano tangente à superfície em cada ponto perpendicular ao gradiente da função no ponto.* □

A hipótese de diferenciabilidade de uma função permite elaborar estratégias de otimização baseadas no fato de que o gradiente de uma função (que, no caso de funções diferenciáveis, é sempre bem definido) indica quais são as direções do espaço para as quais, partindo-se de um ponto, ocorre localmente a diminuição da função. Isso equivale à determinação das direções para as quais se caminha para regiões de sub-nível inferiores. A proposição a seguir formaliza esse fato, que deriva da proposição 3.3.

Proposição 3.4 *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ uma função diferenciável no domínio C, seja x_o um ponto pertencente à superfície de nível $S(f, \alpha)$ e seja $\nabla f(x_o)$ o gradiente de $f(\cdot)$ no ponto x_o. Seja ainda um vetor $d \in \mathbb{R}^n$. Então, se:*

$$d \cdot \nabla f(x_o) < 0 \tag{3.7}$$

então existe $\epsilon > 0$ tal que:

$$f(x_o + \epsilon d) < f(x_o) \tag{3.8}$$

□

3.3 Convexidade e Quasi-Convexidade

Outro tipo de informação que pode ser útil em processos de otimização diz respeito à convexidade das funções.

Definição 3.7 (Função Convexa) *Diz-se que uma função $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ definida sobre um conjunto convexo C é convexa se para quaisquer $x, y \in C$,*

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha) f(y) \tag{3.9}$$

para todo $\alpha \in [0, 1]$. Se para quaisquer $x, y \in C$, sendo $x \neq y$ e $0 < \alpha < 1$, a desigualdade é estrita, então $f(\cdot)$ é estritamente convexa. □

Analogamente, $f(\cdot)$ é (estritamente) côncava se $-f(\cdot)$ for (estritamente) convexa.

Definição 3.8 (Subgradiente) *Seja uma função convexa* $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$. *Um funcional linear* f^{sb} *é um subgradiente de* f *no ponto* x_0 *se:*

$$f(x) \geq f(x_0) + f^{sb}(x - x_0) \quad , \quad \forall\ x \tag{3.10}$$

$\square$

Proposição 3.5 (Caracterizações de Funções Convexas) *Seja* $f(\cdot)$ *uma função duas vezes diferenciável, sobre um conjunto convexo* $C \subset \mathbb{R}^n$. *Então são equivalentes as afirmativas abaixo:*

i. $f\left(\alpha x + (1-\alpha)y\right) \leq \alpha f(x) + (1-\alpha)f(y) \quad \forall\ \alpha \in [0,1]$

ii. $f(y) \geq f(x) + \nabla f(x)'(y - x) \quad \forall\ x, y \in C$

iii. $F(x) \geq 0 \quad \forall\ x \in C$

sendo $\nabla f(x)$ *o vetor gradiente no ponto* x *e* $F(x)$ *a matriz hessiana no ponto* x. $\square$

Exercício 3.1 *Demonstre a proposição 3.5.* $\diamondsuit$

Como no caso de conjuntos convexos, é possível obter funções convexas a partir de combinações convexas.

Proposição 3.6 (Combinações Convexas) *Sejam* $f_i(\cdot) : C_i \subset \mathbb{R}^n \mapsto \mathbb{R}$ *funções convexas definidas sobre conjuntos convexos* C_i , $i = 1, \ldots, m$. *Então:*

i. $\alpha f_i(\cdot)$ *é convexa sobre* C_i, $\forall\ \alpha \geq 0$

ii. $\displaystyle\sum_{i=1}^{m} \alpha_i f_i(\cdot)$ *é convexa sobre* $\displaystyle\bigcap_{i=1}^{m} C_i$ *para* $\alpha_i \geq 0$, $i = 1, \ldots, m$

$\square$

Exercício 3.2 *Demonstre a proposição 3.6.* $\diamondsuit$

Proposição 3.7 *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ uma função convexa sobre C convexo. Então a região de sub-nível $R(f, \alpha)$ é convexa para todo $\alpha \in \mathbb{R}$.* □

A recíproca não é verdadeira.

A convexidade de $R(f, \alpha)$ define um novo tipo de função, as funções *quasi-convexas*:

Definição 3.9 (Função Quasi-Convexa) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ uma função tal que suas regiões de sub-nível $R(f, \alpha)$ são convexas para todo $\alpha \in \mathbb{R}$. Nesse caso, diz-se que $f(\cdot)$ é* quasi-convexa *no domínio C.* □

Proposição 3.8 *Se $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ é uma função quasi-convexa, então:*

$$f(\alpha x + (1 - \alpha)y) \leq \max\{f(x), f(y)\} \quad \forall \;\; x, y \in C \;,\; \forall \; \alpha \in [0, 1] \tag{3.11}$$

□

Outro resultado envolvendo conjuntos e funções convexas pode ser obtido a partir da definição de Epígrafo:

Definição 3.10 (Epígrafo) *O* epígrafo *de uma função $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ é definido como:*

$$[f, C] = \{(x, \theta) \in \mathbb{R}^n \times \mathbb{R} \;:\; x \in C \;,\; f(x) \leq \theta\} \tag{3.12}$$

□

Proposição 3.9 *Uma função $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$ definida sobre C convexo é convexa se e somente se $[f, C]$ é um conjunto convexo.* □

Como todo conjunto convexo, o epígrafo de uma função convexa admite hiperplanos suporte em qualquer ponto de sua fronteira.

A convexidade de funções pode ser relacionada com as regiões de sub-nível, superfícies de nível e bacias de atração.

Proposição 3.10 *Todas as regiões de sub-nível de uma função convexa num domínio convexo são conjuntos convexos.* □

Proposição 3.11 *Uma função convexa em um domínio convexo possui uma única bacia de atração, a qual é um conjunto convexo.* □

A informação a respeito da convexidade de uma função pode ser usada em um processo de otimização a partir da proposição a seguir.

Proposição 3.12 *Seja uma função convexa $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$, seja um ponto qualquer $x_o \in \mathbb{R}^n$, e seja $s(x_o) \in \mathbb{R}^n$ um vetor subgradiente da função no ponto. Então a região de sub-nível que possui o ponto x_o em sua fronteira está contida no semi-espaço fechado negativo definido pelo vetor subgradiente no ponto x_o, ou seja:*

$$\begin{aligned} &E_s = \{x \in \mathbb{R}^n \mid (x - x_o) \cdot s(x_o) \leq 0\} \\ &R(f, f(x_o)) \subset E_s \end{aligned} \tag{3.13}$$

□

3.4 Mínimos Locais e Mínimos Globais

Examina-se agora a questão de definir o que são as soluções do problema de otimização. Uma definição geral de mínimos locais, que abarca todas as possibilidades de sua ocorrência, é apresentada a seguir.

Definição 3.11 (Mínimo Local) *Seja $f(\cdot) : C \subset \mathbb{R}^n \mapsto \mathbb{R}$. Um ponto x^* é um* mínimo local *de $f(\cdot)$ sobre C se existe $\epsilon > 0$ tal que*

$$f(x^*) \leq f(x) \quad , \quad \forall\ x\ \in \mathcal{V}(x^*, \epsilon) \cap C \tag{3.14}$$

onde $\mathcal{V}(x^*, \epsilon) \triangleq \{x : \|x - x^*\| < \epsilon\}$. O ponto $x^* \in C$ é um mínimo local estrito *se vale a desigualdade estrita para $x \neq x^*$.* □

Naturalmente, o conjunto C é o subconjunto do espaço $\mathbb{R}^n$ definido pelas restrições:

$$C \triangleq \{x \in \mathbb{R}^n \mid g_i(x) \leq 0\ ;\ i = 1, \ldots, m\ ;\ h_j(x) = 0\ ;\ j = 1, \ldots, p\} \tag{3.15}$$

É possível, a partir desta definição, construir a definição de mínimo global do funcional. Se for possível escolher $\epsilon > 0$ tal que $\mathcal{V}(x^*, \epsilon) \cap C = C$, então x^* é um mínimo global de $f(\cdot)$ sobre C. O mínimo global é ainda *estrito* se a desigualdade for satisfeita de modo estrito.

3.5 Caracterização dos Mínimos Locais

Considere-se a seguir a notação $\mathcal{C}^n$ para representar o conjunto das funções n vezes continuamente diferenciáveis.

Proposição 3.13 (Otimização Irrestrita) *Seja $f(\cdot) \in \mathcal{C}^2$ e $x^* \in \mathbb{R}^n$. Se forem simultaneamente satisfeitas:*

i. $\nabla f(x^*) = 0$

ii. $F(x^*) > 0$

então x^* é um mínimo local estrito de $f(\cdot)$ sobre $\mathbb{R}^n$. □

Proposição 3.14 (Otimização Restrita - Igualdade) *Sejam $f(\cdot) \in \mathcal{C}^2$ e $h_i(\cdot) \in \mathcal{C}^2$, $i = 1, \ldots, r$ e x^* tal que $h_i(x^*) = 0$, $i = 1, \ldots, r$. Se existem multiplicadores $\lambda_1, \lambda_2, \ldots, \lambda_r$ tais que*

i. $\nabla f(x^*) + \sum_{i=1}^{r} \lambda_i \nabla h_i(x^*) = 0$

ii. $F(x^*) + \sum_{i=1}^{r} \lambda_i H_i(x^*) > 0$ *sobre* $M = \{y \in \mathbb{R}^n \ : \ \nabla h_i(x^*)' y = 0 \ , \ i = 1, \ldots, r\}$

são simultaneamente satisfeitos, então x^* é um mínimo local estrito de $f(\cdot)$ sujeito a $h_i(x) = 0$, $i = 1, \ldots, r$. □

Proposição 3.15 (Otimização Restrita - Desigualdade) *Sejam $f(\cdot) \in \mathcal{C}^2$ e $g_i(\cdot) \in \mathcal{C}^2$, $i = 1, \ldots, p$ e x^* tal que $g_i(x^*) \leq 0$, $i = 1, \ldots, p$. Se existem multiplicadores $\lambda_1, \lambda_2, \ldots, \lambda_p$ tais que*

i. $\lambda_i \geq 0$, $i = 1, \ldots, p$

ii. $\lambda_i g_i(x^*) = 0$, $i = 1, \ldots, p$

iii. $\nabla f(x^*) + \sum_{i=1}^{p} \lambda_i \nabla g_i(x^*) = 0$

iv. $F(x^*) + \sum_{i=1}^{p} \lambda_i G_i(x^*) > 0$ *sobre* $M = \{y \in \mathbb{R}^n : \nabla g_i(x^*)' y = 0 \ , \ i \in I(x^*)\}$,
$I(x^*) = \{i \ : \ g_i(x^*) = 0 \ , \ \lambda_i > 0\}$

são simultaneamente satisfeitos, então x^* é um mínimo local estrito de $f(\cdot)$ sobre $g_i(x) \leq 0$, $i = 1, \ldots, p$. □

Condições necessárias adquirem caráter suficiente se o problema considerado é convexo: $f(\cdot)$ é uma função convexa e as restrições determinam uma região factível convexa. Nestas circunstâncias, qualquer mínimo local de $f(\cdot)$ é também um mínimo global.

Definição 3.12 (Direção Factível) *Seja Ω o conjunto dos pontos factíveis em um problema de otimização com restrições. Diz-se que d é uma direção factível a partir de um ponto $x^0 \in \Omega$ se existe um $\bar{\alpha} > 0$ tal que $(x^0 + \alpha d) \in \Omega\ \forall\ \alpha \in [0, \bar{\alpha}]$.* □

Proposição 3.16 (Condições Necessárias de 1a Ordem) *Seja $\Omega \subset \mathbb{R}^n$ e f uma função diferenciável sobre Ω. Se x^* é um mínimo local de f sobre Ω, então para qualquer $d \in \mathbb{R}^n$ que seja uma direção factível em x^* tem-se que:*

$$\nabla f(x^*)' d \geq 0 \tag{3.16}$$

□

DEMONSTRAÇÃO: Se d é factível em x^*, então

$$x(\alpha) = x^* + \alpha d \quad \in \Omega \ \ \forall\ \alpha \in [0, \bar{\alpha}]$$

Seja $v(\alpha) = f(x(\alpha))$; então v possui um mínimo em $\alpha = 0$. Em série de Taylor:

$$v(\alpha) = v(0) + \dot{v}(0)\alpha + \mathcal{O}(\alpha)$$

Se $\dot{v}(0) < 0$ então para $\alpha > 0$ suficientemente pequeno tem-se $v(\alpha) < v(0)$, o que contraria a hipótese de otimalidade de $v(0)$. Portanto $\dot{v}(0) = \nabla f(x)' d \geq 0$. ■

Proposição 3.17 (Condições Necessárias de 2a Ordem) *Seja $\Omega \subset \mathbb{R}^n$ e f uma função duas vezes diferenciável sobre Ω. Se x^* é um mínimo local de f sobre Ω, então para qualquer $d \in \mathbb{R}^n$ factível em x^*, tem-se que:*

i. $\nabla f(x^*)' d \geq 0$

ii. Se $\nabla f(x^*)' d = 0$, *então* $d' F(x^*) d \geq 0$

□

DEMONSTRAÇÃO: Suponha que $\dot{v}(0) = \nabla f(x^*)' d = 0$. Neste caso:

$$v(\alpha) = v(0) + \frac{1}{2}\ddot{v}(0)\alpha^2 + \mathcal{O}(\alpha^2)$$

Se $\ddot{v}(0) < 0$, então $v(\alpha) < v(0)$ para $\alpha > 0$ suficientemente pequeno, o que contraria a hipótese de otimalidade. Portanto $\ddot{v}(0) = d'F(x^*)d \geq 0$. ■

NOTA 3.2 *Observe-se que se x^* é um ponto interior de Ω então as condições se reduzem a:*

i. $\nabla f(x^*) = 0$

ii. $d'F(x^*)d \geq 0 \quad , \quad \forall\, d \in \mathbb{R}^n$

◇

Definição 3.13 (Ponto Regular) *Um ponto x^* satisfazendo um conjunto de restrições de igualdade $h(x^*) = 0$ e um conjunto de restrições de desigualdade $g(x^*) \leq 0$ é chamado de* ponto regular *dessas restrições se os vetores gradiente $\nabla h_i(x^*)$ e $\nabla g_j(x^*)$ para todo j tal que $g_j(x^*) = 0$ forem linearmente independentes.* □

A seguinte condição de primeira ordem para otimalidade de um problema genérico de otimização, apresentada por Kuhn e Tucker em 1951, possui importância histórica, servindo de base para diversos algoritmos numéricos de otimização existentes.

Proposição 3.18 (Condições Necessárias de Kuhn-Tucker para Otimalidade) *Seja x^* um ponto regular das restrições do problema de otimização:*

$$\min \ f(x)$$

$$\textit{sujeito a} \begin{cases} h_i(x) = 0 \ , \ i = 1, \ldots, l \\ \\ g_j(x) \leq 0 \ , \ j = 1, \ldots, m \end{cases} \tag{3.17}$$

sendo que f, g, $h \in \mathcal{C}^1$. Para x^* ser um ótimo local do problema, deve existir um conjunto de multiplicadores de Kuhn-Tucker $\lambda^* \in \mathbb{R}^l$ e $\mu^* \in \mathbb{R}^m$ com $\mu^* \geq 0$ tal que:

$$
\begin{aligned}
&\nabla f(x^*) + \sum_{i=1}^{l} \lambda_i^* \nabla h_i(x^*) + \sum_{j=1}^{m} \mu_j^* \nabla g_j(x^*) = 0 \\
&\mu_j^* g_j(x^*) = 0 \quad \forall\ j = 1, \ldots, m
\end{aligned}
\tag{3.18}
$$

□

NOTA 3.3 *Essas condições necessárias tornam-se também suficientes em problemas convexos (com função objetivo convexa e todas as restrições convexas).*

♢

EXEMPLO 3.1 *Considere-se o problema de otimização definido por:*

$$x^* = \arg\ \min_x\ f(x)$$

$$\textit{sujeito a:} \begin{cases} g_1(x) \leq 0 \\ \\ g_2(x) \leq 0 \end{cases}$$

sendo:

$$f(x) = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$g_1(x) = 1 - x_1$$

$$g_2(x) = 1 - x_2$$

A solução desse problema é o ponto:

$$x^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Pode-se verificar que esse ponto atende às condições de Kuhn-Tucker, pois:

$$\nabla f(x^*) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \qquad \nabla g_1(x^*) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad \nabla g_2(x^*) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

de forma que:

$$\nabla f(x^*) + 2\nabla g_1(x^*) + 2\nabla g_2(x^*) = 0$$

Deve-se notar que, como a função objetivo e as restrições desse problema são todos convexos, as condições de Kuhn-Tucker tornam-se necessárias e suficientes para caracterizar a otimalidade da solução. ♢

EXEMPLO 3.2 *Considere-se o problema de otimização definido por:*

$$x^* = \arg\ \min_x\ f(x)$$

$$\textit{sujeito a: } \left\{\ g(x) \leq 0\right.$$

sendo:

$$f(x) = \begin{bmatrix} x_1 - 1 & x_2 - 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

$$g(x) = 8 - \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A solução desse problema é o ponto:

$$x^* = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Pode-se verificar que esse ponto atende às condições de Kuhn-Tucker, pois:

$$\nabla f(x^*) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \qquad \nabla g(x^*) = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

de forma que:

$$2\nabla f(x^*) + \nabla g(x^*) = 0$$

Este problema possui função objetivo convexa, porém a restrição é uma função côncava. Dessa forma, as condições de Kuhn-Tucker são apenas necessárias para a otimalidade local de um ponto, sendo possível que haja pontos que atendam às condições e não sejam mínimos locais do problema. Esse é o caso do ponto:

$$\bar{x} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

Nesse ponto observa-se que:

$$\nabla f(\bar{x}) = \begin{bmatrix} -6 \\ -6 \end{bmatrix} \qquad \nabla g(\bar{x}) = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

de forma que:

$$\frac{1}{3}\nabla f(x^*) + \frac{1}{2}\nabla g(x^*) = 0$$

o que significa que as condições de Kuhn-Tucker se verificam no ponto. Entretanto, o ponto $\bar{x}$ não é um mínimo local do problema, uma vez que há pontos com valores menores de função objetivo em toda vizinhança que se tomar do mesmo. ♢

Capítulo 4

Convergência de Algoritmos

Um aspecto importante na construção de algoritmos de otimização é a determinação de condições mediante as quais tais algoritmos convergem para as soluções do problema.

Este capítulo irá lidar com dois conceitos distintos porém relacionados com o problema da convergência: Primeiro, será abordada a questão de estabelecer que um algoritmo *converge* para a solução do problema. A seguir, será discutida a *velocidade* com que ocorre tal convergência.

A maior parte do material deste capítulo foi extraída da referência (Luenberger 1984).

4.1 Algoritmos

Um algoritmo é definido como um mapeamento recursivo. A forma mais simples de definir algoritmos é baseada no conceito de mapeamento de um ponto para um ponto:

Definição 4.1 (Algoritmo) *Seja $x_0 \in \mathbb{R}^n$, e seja o operador $A(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^n$. Esse operador é denominado* algoritmo *e a seqüência definida por:*

$$x_{k+1} = A(x_k) \tag{4.1}$$

a partir de x_0 é denominada saída *do algoritmo para a entrada x_0.* □

Uma outra definição de algoritmo que também se faz útil para a análise de determinados problemas é mais geral, e se baseia no conceito de mapeamento de um ponto para um conjunto.

Definição 4.2 (Algoritmo Generalizado) *Seja o espaço vetorial* $\mathbb{R}^n$. *Defina-se* $\mathcal{Q}$ *como o conjunto de todos os subconjuntos desse espaço. Seja o operador* $A(\cdot) : \mathbb{R}^n \mapsto \mathcal{Q}$ *tal que:*

$$\mathcal{X} = A(x) \tag{4.2}$$

sendo $x \in \mathbb{R}^n$ *e* $\mathcal{X} \in \mathcal{Q}$. *Esse operador é um* algoritmo generalizado. □

Definição 4.3 (Saída do Algoritmo Generalizado) *Toda seqüência* $[x_k]$ *gerada segundo a lei:*

$$\begin{aligned} \mathcal{X}_{i+1} &= A(x_i) \\ x_{i+1} &\in \mathcal{X}_{i+1} \end{aligned} \tag{4.3}$$

ou seja, tomando como x_{k+1} *qualquer ponto pertencente ao conjunto* $\mathcal{X}_{k+1}$ *é chamada de* saída do algoritmo generalizado, *para um dado ponto inicial* x_0. □

Um algoritmo generalizado, claramente, inclui algum grau de incerteza que não está presente no algoritmo ordinário. Isso pode ser importante para refletir as condições sob as quais operam algoritmos de otimização efetivamente implementados. Evidentemente, a convergência de um algoritmo generalizado para as soluções de um problema deve ser estabelecida na forma da convegência do conjunto $\mathcal{X}_k$ para o conjunto-solução.

A seguir, são estabelecidos conceitos em que se apoiará a construção de um teorema de convergência de algoritmos generalizados.

Definição 4.4 (Função Descendente) *Seja um algoritmo generalizado* $A(\cdot)$ *no espaço* $\mathbb{R}^n$. *Seja* $\Gamma \subset \mathbb{R}^n$ *um conjunto denominado* conjunto solução, *e seja* $z(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$. *A função* $z(\cdot)$ *é dita* função descendente *para* Γ *e* A *se satisfizer:*

i. $x \notin \Gamma \ , \ y \in A(x) \ \Rightarrow \ z(y) < z(x)$

ii. $x \in \Gamma \ , \ y \in A(x) \ \Rightarrow \ z(y) \leq z(x)$

□

Definição 4.5 (Algoritmo Fechado) *Um algoritmo generalizado* A *no espaço* $\mathbb{R}^n$ *é dito* fechado *no ponto* x *se as premissas:*

i. $x_k \to x$, $x_k \in \mathbb{R}^n$

ii. $y_k \to y$, $y_k \in A(x_k)$

implicarem que:

$$y \in A(x)$$

O algoritmo generalizado A é dito fechado num domínio D se for fechado em todos os pontos de D. $\square$

Deve-se observar que o conceito de *algoritmo fechado*, aplicável a algoritmos generalizados, recai no conceito de *mapeamento contínuo*, quando se particularizam as condições para o caso de algoritmos ordinários. Ou seja, um algoritmo ordinário (ponto-para-ponto) que for um mapeamento contínuo será um caso particular de algoritmo fechado. Com esses conceitos, é possível estabelecer um teorema que estabelece condições suficientes para a convergência global de algoritmos fechados.

Teorema 4.1 (Teorema da Convergência Global) *Seja A um algoritmo fechado num espaço $\mathbb{R}^n$. Suponha-se que, dado um ponto $x_0 \in \mathbb{R}^n$, a seqüência $[x_k]_{k=0}^{\infty}$ é gerada, satisfazendo*

$$x_{k+1} \in A(x_k)$$

Seja dado um conjunto-solução $\Gamma \subset \mathbb{R}^n$, e suponha-se que:

i. Todos os pontos x_k estão contidos num conjunto compacto $S \subset \mathbb{R}^n$;

ii. Existe uma função contínua $z(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$ tal que:

(a) $x \notin \Gamma \;\Rightarrow\; z(y) < z(x) \;\forall\; y \in A(x)$

(b) $x \in \Gamma \;\Rightarrow\; z(y) \leq z(x) \;\forall\; y \in A(x)$

iii. O algoritmo A é fechado em todos os pontos fora de Γ.

Então o limite de toda subseqüência convergente $[x_k]$ é uma solução (pertence a Γ). $\square$

DEMONSTRAÇÃO: Suponha-se uma subseqüência convergente $[x_k], k \in \mathcal{K}$ que converge para o limite x. Uma vez que $z(\cdot)$ é uma função contínua, segue-se que para $k \in \mathcal{K}$, tem-se que $z(x_k) \to z(x)$. Isto significa que $z(\cdot)$ é convergente em relação a esta subseqüência. Pela monotonicidade de $z(\cdot)$ na seqüência $[x_k]$ tem-se que $z(x_k) - z(x) > 0$ para todo k. Pela convergência de $z(\cdot)$ na subseqüência, existe, para um certo $\epsilon > 0$, algum $K \in \mathcal{K}$ tal que $z(x_k) - z(x) < \epsilon$ para todo $k > K$, com $k \in \mathcal{K}$. Assim, para todo $k > K$:

$$z(x_k) - z(x) = z(x_k) - z(x_K) + z(x_K) - z(x) < \epsilon$$

Isso estabelece que $z(x_k) \to z(x)$.

Para completar a prova falta mostrar que x é uma solução. Suponha-se que não seja. Considere-se a subseqüência $[x_{k+1}]_{\mathcal{K}}$. Uma vez que todos os pontos dessa seqüência estão contidos em um conjunto compacto, existe um $\bar{\mathcal{K}} \subset \mathcal{K}$ tal que $[x_{k+1}]_{\mathcal{K}}$ converge para algum limite $\bar{x}$. Tem-se então que $x_k \to x$, $k \in \bar{\mathcal{K}}$, e $x_{k+1} \in A(x_k)$ com $x_{k+1} \to \bar{x}$, $k \in \mathcal{K}$. Assim, uma vez que A é fechado em x segue-se que $\bar{x} \in A(x)$. Mas isso implica $z(\bar{x}) = z(x)$, o que contradiz o fato de que $z(\cdot)$ é uma função descendente. $\blacksquare$

Corolário 4.1 *Sob as condições do teorema 4.1, se Γ consiste de um único ponto x^*, então toda seqüência $[x_k]$ converge para x^*.* $\square$

Bibliografia

Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. (1985). A learning algorithm for Boltzman Machines, *Cognitive Science* **9**: 147–169.

Akgül, M. (1984). *Topics in Relaxation and Ellipsoidal Methods*, number 97 in *Research Notes in Mathematics*, Pitman Publishing Inc., London, UK.

Belmont-Moreno, E. (2001). The role of mutation and population size in genetic algorithms applied to physics problems, *International Journal of Modern Physics C* **12**(9): 1345–1355.

Bland, R. G., Goldfarb, D. & Todd, M. J. (1981). The ellipsoid method: a survey, *Operations Research* **29**(6): 1039–1091.

Cao, Y.-Y., Lam, J. & Sun, Y.-X. (1998). Static output feedback stabilization: an ILMI approach, *Automatica* **34**(12): 1641–1645.

Chankong, V. & Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*, North-Holland (Elsevier), New York.

Chen, B. S., Cheng, Y. M. & Lee, C. H. (1995). A genetic approach to mixed $\mathcal{H}_\infty/\mathcal{H}_2$ optimal PID control, *IEEE Control Systems Magazine* **15**(5): 51–60.

Chen, C. T. (1984). *Linear System Theory and Design*, Hartcourt Brace College Pub.

Choi, D. H. & Oh, S. Y. (2000). A new mutation rule for evolutionary programming motivated from backpropagation learning, *IEEE Transactions on Evolutionary Computation* **4**(2): 188–190.

Dias-Filho, W. (2003). *Algoritmos Cone-Elipsoidais para Geração de Soluções Eficientes: Construção e Aplicações*, PhD thesis, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, Belo Horizonte, MG.

Dorigo, M., Maniezzo, V. & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Sys. Man Cyb. - Part B* **26**(1): 29–41.

Dziuban, S. T., Ecker, J. G. & Kupferschmid, M. (1985). Using deep cuts in an ellipsoidal algorithm for nonlinear programming, *Math. Program. Study* **25**: 93–107.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

J. C. Potts, T. D. G. & Yadav, S. B. (1994). The development and evaluation of an improved genetic algorithm based on migration and artificial selection, *IEEE Transactions on Systems, Man and Cybernetics* **24**(1): 73–86.

Jain, A. & Zongker, D. (1997). Feature selection: evaluation, application, and small sample performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(2): 153–158.

Johnson, J. M. & Ramat-Semii, Y. (1997). Genetic algorithms in engineering electromagnetics, *IEEE Antennas and Propagation Magazine* **39**(4): 7–25.

K. F. Man, K. S. T. & Kwong, S. (1996). Genetic algorithms: concepts and applications, *IEEE Transactions on Industrial Electronics* **43**(5): 519–534.

K. Rashid, J. A. R. & Freeman, E. M. (2000). A general approach for extracting sensitivity analysis from a neuro-fuzzy model, *IEEE Transactions on Magnetics* **36**(4): 1066–1070.

Khargonekar, P. P. & Rotea, M. A. (1991). Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: a convex optimization approach, *IEEE Transactions on Automatic Control* **36**(7): 824–837.

Luenberger, D. G. (1984). *Linear and Nonlinear Programming*, Addison-Wesley.

Meneguim, R. A. (1999). *Análise da sensibilidade de soluções em otimização através de elipsóides mínimos*, Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, Belo Horizonte, MG.

Miranda, M. F. (2000). *Controle Multivariável na Presença de Incertezas*, PhD thesis, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, Belo Horizonte, MG.

Naudts, B. & Kallel, L. (2000). A comparison of predictive measures of problem difficulty in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* **4**(1): 1–15.

Popper, K. R. (1974). *A Lógica da Pesquisa Científica (trad.)*, Ed. Cultrix. (Logic der Forschung, 5a. ed. 1973; 1a. ed. 1934).

Rudin, W. (1991). *Functional Analysis*, McGraw-Hill.

Saldanha, R. R., Takahashi, R. H. C., Vasconcelos, J. A. & Ramirez, J. A. (1999). Adaptive deep-cut method in ellipsoidal optimization for electromagnetic design, *IEEE Transactions on Magnetics, Part I* **35**(3): 1746–1749.

Sareni, B. & Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited, *IEEE Transactions on Evolutionary Computation* **2**(3): 97–106.

Schell, T. & Wegenkittl, S. (2001). Looking beyond selection probabilities: adaptation of the χ^2 measure for the performance analysis of selection methods in GA's, *Evolutionary Computation* **9**(2): 243–256.

Scherer, C., Gahinet, P. & Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization, *IEEE Transactions on Automatic Control* **42**(7): 896–911.

Shah, S., Mitchell, J. E. & Kupferschmid, M. (2001). An ellipsoid algorithm for equality-constrained nonlinear programs, *Computers and Operations Research* **28**: 85–92.

Shimomura, T. & Fujii, T. (2000). Multiobjective control design via sucessive over-bounding of quadratic terms, *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 2763–2768.

Takahashi, R. H. C., Palhares, R. M., Dutra, D. A. & Gonçalves, L. P. S. (2001). Synthesis and characterization of Pareto-optimal solutions for the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problem, *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, USA, pp. 3997–4002.

Takahashi, R. H. C., Peres, P. L. D. & Ferreira, P. A. V. (1997). Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ guaranteed cost PID design, *IEEE Control Systems Magazine* **17**(5): 37–47.

Takahashi, R. H. C., Ramirez, J. A., Vasconcelos, J. A. & Saldanha, R. R. (2001). Sensitivity analysis for optimization problems solved by stochastic methods, *IEEE Transactions on Magnetics* **37**(5): 3414–3417.

Takahashi, R. H. C., Saldanha, R. R., Dias-Filho, W. & Ramirez, J. A. (2003). A new constrained ellipsoidal algorithm for nonlinear optimization with equality constraints, *IEEE Transactions on Magnetics* **39**(3): 1289–1292.

Takahashi, R. H. C., Vasconcelos, J. A., Ramirez, J. A. & Krahenbuhl, L. (2003). A multiobjective methodology for evaluating genetic operators, *IEEE Transactions on Magnetics* **39**(3): 1321–1324.

Tanomaru, J. (1995). Motivação, fundamentos e aplicações de algoritmos genéticos, *Anais do II Congr. Bras. de Redes Neurais*, Vol. 1, Curitiba, PR, Brasil.

Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C. & Saldanha, R. R. (2001). Improvements in genetic algorithms, *IEEE Transactions on Magnetics* **37**(5): 3414–3417.

Vidyasagar, M. (1993). *Nonlinear Systems Analysis*, Prentice-Hall.

Viennet, R., Fonteix, C. & Marc, I. (1996). Multicriteria optimization using a genetic algorithm for determining a Pareto set, *Int. J. Sys. Sci.* **27**(2): 255–260.

Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* **1**(1): 67–82.

Z. Michalewicz, K. Deb, M. S. & Stidsen, T. (2000). Test-case generator for nonlinear continuous parameter optimization techniques, *IEEE Transactions on Evolutionary Computation* **4**(3): 197–215.