

CAPÍTULO 1: INTRODUÇÃO - O PROBLEMA ESTUDADO

I. Motivação - o problema de otimização da unidade FCC

O conversor de craqueamento catalítico em leito fluidizado (FCC) constante da unidade FCC é um exemplo desafiador típico do problema de controle da engenharia química (Mac Farlane & Reinemann, 1990). A função do conversor FCC nas refinarias é converter a corrente de gásóleo oriunda do petróleo em correntes de hidrocarbonetos de cadeias menores. Destes a gasolina é um dos produtos mais valiosos. O craqueamento no conversor FCC é considerado o maior processo catalítico do mundo em função das dimensões dos equipamentos envolvidos, volumes processados e quantidade de catalisador usada (Moro, 1992). Outrossim, trata-se de um sistema multivariável, altamente não linear, com fortes interações entre as variáveis e em que existem restrições nas variáveis de entrada e saída (Lopez et al., 1996, Odloak & Gouvêa, 1996). Todas estas características fazem com que o conversor FCC seja um sistema de difícil controle.

A grande quantidade de material processado e a capacidade de produzir gasolina fazem com que a unidade FCC tenha uma importância acentuada para a rentabilidade da refinaria. Isto pode ser visualizado no fato de que o lucro líquido diário decorrente da unidade FCC é da ordem de 30 dólares por tonelada de carga processada e que uma unidade típica processa 8000 ton/dia (Moro, 1992). Assim, a implementação de estratégias de controle avançado e o esforço em se realizar uma otimização do processo são justificados. Na literatura, diversos trabalhos podem ser encontrados a respeito da unidade FCC (Ali & Elnashaie, 1997, Lona Batista & Maciel Filho, 1997, Campos et al., 1997, Pilia et al., 1997, Den Hollander et al., 1997, Knops Gerrits et al., 1997, Arbel et al., 1997, 1996, 1995a, 1995b, Abou-Jeyab & Gupta, 1996, Lopez et al., 1996, Odloak & Gouvêa, 1996, Huq et al., 1995, Moro & Odloak, 1995; Mc Farlane & Reineman, 1993, 1990; Grosdidier et al., 1993; Rhemann et al., 1989, Prett & Gillette, 1980). Apesar do grande número de trabalhos publicados relativos ao conversor FCC, e muitos deles recentes, o enfoque maior

tem sido com relação ao estabelecimento de um modelo adequado e formas de controlar o processo. O problema de otimização permanece ainda negligenciado. É pois de interesse que se avalie como a otimização deste processo pode ser feita.

Nos parágrafos precedentes, colocamos o interesse da otimização em termos econômicos. Ao mesmo tempo, ressaltamos a dificuldade de se controlar adequadamente o processo. Assim, intui-se que o problema da otimização é dependente do problema de controle e que o primeiro não pode ser considerado sem o segundo. Por outro lado, o objetivo de se resolver o problema da otimização da unidade FCC pode ser generalizado para o interesse de se efetuar a otimização de processos químicos contínuos, sendo o conversor FCC um exemplo complexo destes últimos. Nesta visão, procura-se entender o que vem a ser o problema de otimização de processos químicos contínuos, o que se deve fazer para resolvê-lo e buscar as ferramentas necessárias para tal fim. Em linhas gerais, esta é a motivação da presente tese, ou seja, exemplificar o problema de otimização de processos químicos contínuos e estabelecer as ferramentas necessárias para a sua resolução.

Para que se entenda o problema a ser estudado, no item II, caracterizamos o ambiente industrial e com isto o problema de otimização em tempo real é definido e descrito com pormenores. Em seguida, no item III, discutimos o que é estudado na presente tese, mostrando os objetivos propostos e as inovações feitas. Finalmente, no item IV, comentamos sobre a forma de apresentação da presente tese.

II. O ambiente industrial e a necessidade da otimização - o problema de otimização em tempo real

A operação de uma planta química deve ser feita de forma a satisfazer diversos requisitos mesmo na presença de perturbações (Kwong, 1993). Estes requisitos englobam questões ambientais, econômicas, de segurança, de especificação de produtos e restrições operacionais, as quais podemos dividir em dois grupos básicos. O primeiro corresponde a uma regulação de certas variáveis de tal forma que as especificações de produtos e restrições operacionais sejam atendidas. Como ocorrem variações na planta, intervenções externas são necessárias. Trata-se do problema de *controle do processo*. O segundo grupo de objetivos a serem atendidos é proveniente de considerações econômicas. Neste, deseja-se, tipicamente, maximizar o lucro ou minimizar os custos de operação. Trata-se do problema de *otimização das condições operacionais*, o qual é resolvido de tempos em tempos de acordo com a necessidade existente. A otimização pode ser feita dentro de uma estrutura hierárquica de controle, sendo, então, denominada de *otimização em tempo real*. Os objetivos do primeiro grupo sempre devem ser obedecidos e a procura do ponto ótimo de operação (para que se tenha máximo lucro ou mínimo custo) não pode violá-los. O uso de uma *estratégia de otimização em tempo real* visa a se atender estes objetivos e pode ser, inicialmente, entendida como um procedimento constituído das seguintes duas etapas:

- *identificação da necessidade de se alterar o ponto de operação da planta de forma a se minimizar o custo operacional*: diversas variáveis da planta devem ser continuamente monitoradas, as quais devem refletir perturbações no processo que alterem as variáveis econômicas de interesse. Ainda, os modelos eventualmente empregados para representar o processo precisam ser atualizados sempre que mudanças significativas no ponto de operação da planta são realizadas.
- *geração de valores de referência ou cálculo das novas variáveis manipuladas que atendam ao novo ponto de operação ótimo*: no primeiro caso, denomina-se a estratégia de otimização em duas camadas e no segundo caso de otimização em uma camada. A figura 1.I.1 mostra esquematicamente como cada uma delas é realizada. Basicamente, no primeiro caso, a resolução do problema econômico é feita isoladamente e a solução é enviada ao controlador, que atua no processo, garantindo a operação estável e segura da planta. Em cada uma das camadas, i.e., a de otimização e controle, faz-se uso de um modelo do processo, o qual é atualizado através da estimação de certos parâmetros deste. Existem perturbações nas plantas, as quais não são medidas. Ainda, algumas variáveis também não o são, sendo então estimadas. No segundo caso, o controlador incorpora o cálculo da função econômica. O modelo usado é único e contém informações econômicas e do

processo. Este modelo deve ser como no caso anterior atualizado de tempos em tempos. Nesta situação, o custo computacional para se resolver o problema pode tornar-se proibitivo se a dimensão do problema for elevada.

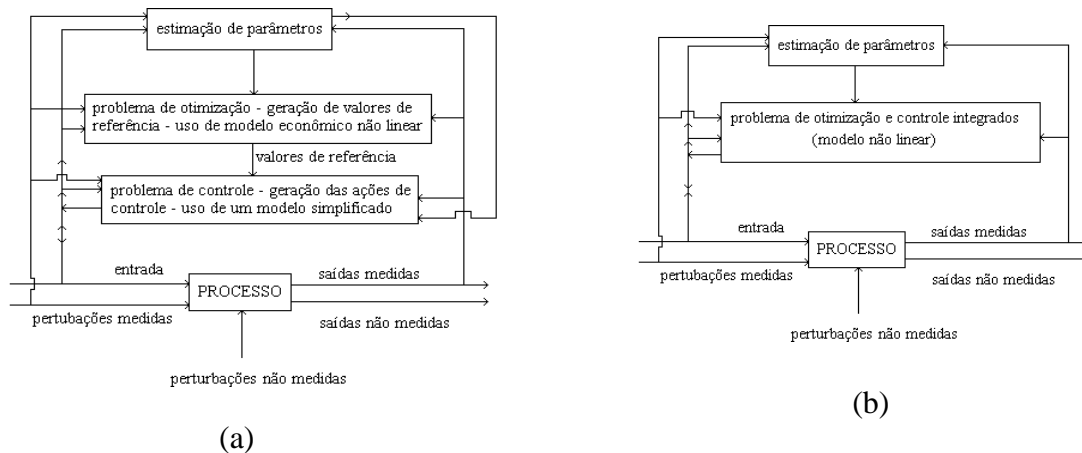


figura 1.II.1: esquema de otimização: (a) em duas camadas (b) em uma camada

De acordo com Cutler & Perry (1983) pode-se conseguir um acréscimo na receita de um dado processo da ordem de 6-10% com o uso da otimização. Para uma planta de etileno com capacidade de 1 bilhão lb/ano, um aumento de 1% no rendimento do processo pode significar um ganho econômico da ordem de US\$ 1.500.000/ano. Segundo Poje & Smart (1986) os custos da implementação de um sistema computacional completo para a otimização pode-se situar na faixa de US\$ 750.000. Hoje em dia, estes custos são menores e outros decorrentes de *softwares* e custos de homem - hora tornam-se significativos. Percebemos desta forma que o retorno do investimento é rápido. Assim, é grande o interesse em se estudar a otimização de processos (Cutler, 1993, Biegler, 1993, Vasantharajan et ali, 1990, Renfro et ali, 1987). Este é ainda mais acentuado visto à crescente necessidade de se assegurar a competitividade dos processos industriais, como pode ser notado nos trabalhos publicados. Por exemplo, Rhemann et ali (1989) relatam a implementação da otimização e controle avançado em uma unidade de craqueamento catalítico em leito fluidizado (FCC) da companhia OEMV na Áustria.

Durante o procedimento de solução do problema de otimização em tempo real, diversos obstáculos são encontrados, os quais se devem às dificuldades inerentes dos componentes constituintes deste. Darby & White (1988) descrevem os elementos do problema de otimização em tempo real como sendo:

- *modelo do processo*: o modelo deve representar o comportamento do processo e pode incorporar a influência de perturbações, degradação dos equipamentos, especificações dos produtos, etc. O modelo é constituído normalmente de relações

físico-químicas, sendo portanto não linear. É a partir dele que os valores de referência que caracterizarão o novo ponto de operação ótimo podem ser gerados para o controlador.

- *função objetivo*: tipicamente, deseja-se maximizar o lucro ou minimizar o custo operacional através de uma manipulação de certas variáveis do processo, a qual atenda as restrições existentes. Nestes casos, a função objetivo é usualmente da forma:

$$\text{objetivo} = \text{valor dos produtos} - \text{custo das matérias primas} - \text{custo de utilidades} + \text{outros efeitos econômicos}$$

Alternativamente, o objetivo pode ser o de aumentar a produção de determinado item, melhorar a qualidade ou alterar a característica de um produto, objetivos estes que podem ser alternados ao longo do ano.

O estabelecimento da função econômica pode não ser trivial e depende de propriedades dos produtos ou da qualidade das matérias-primas empregadas.

- *algoritmo de otimização*: o algoritmo de otimização é um método numérico que usa o modelo e a função objetivo para resolver o problema econômico e/ou de controle. Tipicamente tem-se um problema cujas equações são bastante complexas ou numerosas. Existem inúmeros algoritmos, a escolha do ideal deve levar em conta o esforço computacional requerido e a robustez deste.
- *validação dos dados da entrada*: alguns problemas relacionados à validação dos dados podem ser citados, tais como: escolha do período de amostragem, forma de calcular valores médios, testes de verificação do estado estacionário, forma de inferir determinadas variáveis, testes de consistência dos dados, reconciliação de dados.
- *implementação dinâmica*: no caso da otimização em duas camadas, uma vez, calculadas as variáveis que determinam o ponto ótimo de operação, estas são passadas para o controlador que toma as ações devidas de forma que as restrições do processo sejam atendidas. O ponto de operação real irá diferir do ponto otimizado devido a erros existentes no modelo. No caso da otimização em uma camada, o que temos é que o problema de otimização dificulta o de controle, uma vez que a sintonia deste último, de forma que problemas de estabilidade não surjam, pode não ser um procedimento trivial.
- *atualização do modelo*: os modelos empregados são apenas descrições do processo real e desta forma existem erros em relação a este. Ainda, determinados parâmetros podem variar com o tempo devido a inúmeros fatores como por exemplo problemas de incrustação, etc. Novas restrições podem ou não ser incorporadas. Logo a atualização do modelo é essencial.

As dificuldades de resolução do problema de otimização em tempo real podem então ser expressas como:

- **estabelecimento do modelo matemático:** as dificuldades neste item são as habitualmente conhecidas, i.e., se devem ao compromisso de se ter um modelo capaz de reproduzir os dados da planta e de ser este relativamente simples. Adicionalmente, deve ser possível estabelecer uma função econômica que tenha como parâmetros dados mensuráveis. O problema econômico resultante que se quer resolver é constituído de funções não lineares e recebe o nome de problema da programação não linear (PPNL).
- **uso de um algoritmo de solução adequado:** basicamente, o que devemos ter é um método numérico que seja capaz de resolver o PPNL de uma forma eficiente e robusta. O que ocorre é que ainda não existe um algoritmo que tenha estas duas características desejadas para todos os problemas da PPNL. Assim, duas alternativas surgem, usar os algoritmos existentes quando isto é possível ou desenvolver um novo, sendo a última alternativa uma área de pesquisa em aberto.
- **validação dos dados de entrada:** todas as medidas que são lidas devem ser validadas. Sucintamente, a resolução desta etapa pode ser feita através da técnica de reconciliação de dados. Ainda, dependendo de como a estratégia de otimização é formulada, é necessário saber identificar se a planta está no estado estacionário. Este procedimento pode não ser trivial.
- **implementação dinâmica:** esta etapa está relacionada com a garantia de se ter uma estratégia robusta. Basicamente, as ações efetivamente implementadas na planta não podem desestabilizá-la. Pode-se querer fazer uso da teoria de controle robusto para se minimizar efeitos de perturbações não medidas e erros nos modelos usados. Este procedimento não é trivial e dependendo de como a estratégia de otimização é formulada, podem não existir resultados da literatura que possam ser aplicados. Ou seja, esta também é uma área de pesquisa em aberto.
- **validação do modelo:** como o modelo é uma simplificação do processo real, os parâmetros deste devem ser continuamente atualizados, procedimento este não trivial, especialmente para o caso não linear. Novamente, trata-se de uma área de pesquisa em aberto.

Percebemos que o assunto da otimização de um processo em tempo real é muito complexo. O que ocorre é que cada pesquisador atuando nesta área contribui um pouco para a resolução de cada um destes itens. Verificamos que um dos itens mais críticos para que se possa realizar a otimização em tempo real é se ter à disponibilidade um algoritmo de resolução dos problemas da PNL robusto e eficiente. A questão da **robustez**, entendida na presente tese, **como a capacidade do algoritmo em obter uma solução estacionária** e portanto viável para todo problema possível que se quer resolver é um quesito que freqüentemente não se observa nos algoritmos da literatura. Esta característica é de vital importância, uma vez, que na otimização em tempo real, atua-se constantemente no processo, o qual não pode ficar sem controle, situação que pode corresponder ao caso de não se obter uma solução viável. A obtenção de uma solução de cota ou de máximo ao invés de mínimo é menos crítica sob o ponto de vista de operação da planta. Neste caso, pode-se, e.g., desconsiderar os valores de referência gerados na otimização ou incluir restrições na própria função objetivo.

A não existência de um algoritmo robusto, em outras palavras, confiável, invalida qualquer estratégia de otimização, isto é, nada garante que ela irá de fato funcionar. Por esta razão concentramos a nossa atenção, na presente tese, no estabelecimento de um algoritmo mais robusto em relação aos existentes. Ainda, mostramos o seu uso para a resolução do problema de otimização em tempo real do conversor FCC. Propusemos algumas formas de efetuar a implementação da otimização e assim contribuímos em dois dos acima mencionados itens para a resolução do problema da otimização. As contribuições são melhor caracterizadas no item seguinte.

III. Estudos realizados e relevância destes - objetivos e inovação

No item anterior, destacamos o problema da otimização em tempo real e mostramos quais são os assuntos que a compõem. Escolhemos como alvo de pesquisa dois destes itens, a saber: *a elaboração de um algoritmo de solução dos problemas da PNL e a propositura de uma estratégia de otimização*, sendo estes os dois objetivos básicos da presente tese. Com isto, concluímos que a realização da otimização em tempo real é factível nos processos industriais. Com relação ao esforço computacional gasto, verificamos que talvez possa haver a restrição de não se poder incluir muitos equipamentos de uma só vez no problema econômico, mas que a aplicação da estratégia por pedaços da planta é realizável.

Diversas inovações foram introduzidas na presente tese, as quais se dividem nos seguintes quatro aspectos básicos: modificações no algoritmo de programação quadrática sucessiva (SQP) clássico, classificação e resolução de problemas da PNL, caracterização das soluções do problema da PQ, análise do procedimento da otimização em tempo real. Na discussão que segue iremos tratar de cada um destes itens.

Com relação à elaboração do algoritmo SQP, muitas inovações foram introduzidas, especialmente no que tange aos resultados obtidos. A principal contribuição cremos que se deu com relação à flexibilidade do algoritmo que se obteve com a introdução de dois parâmetros de sintonia, inexistentes nos algoritmos da literatura. Inicialmente, na tese, efetuamos uma discussão crítica sobre os algoritmos existentes e dos problemas que podem surgir. A partir daí, verificamos o interesse em se trabalhar com problemas da programação quadrática (PQ) não convexos e desenvolvemos um algoritmo para tal fim. Assim, o Hessiano é calculado por diferenças finitas ou por expressões analíticas. Neste ponto, cabe salientar, que de nosso conhecimento existem apenas duas outras linhas de pesquisa na literatura com relação a se trabalhar com problemas da PQ não convexos, as de Bartholomew-Biggs & Hernandez (1995) e de Lucia et al (1996). As inovações do algoritmo se resumem aos seguintes itens:

- *tratamento para os casos em que as restrições de limites são linearmente dependentes das de igualdade*: a direção de Newton gerada é confinada de forma que restrições de limite nas variáveis não sejam violadas.

- *tratamento para o caso em que os gradientes das restrições de igualdade são nulos e para o caso em que o gradiente reduzido das restrições de desigualdade é nulo:* tratamentos simplificados são mostrados. O procedimento mostrou ser eficiente, embora para alguns dos tratamentos não exista garantia de convergência global. As alterações para este caso, no entanto, podem ser facilmente incorporadas.
- *forma de se atualizar e preparar o conjunto ativo:* permite-se que se ative uma restrição à mais em relação ao número de variáveis do problema da PNL. Desta forma, introduz-se um novo tratamento de restrições linearmente dependentes. A estrutura permite que problemas da PQ inviáveis ou bastante não convexos, os quais denominamos de críticos ou ardilosos sejam identificados. Com isto dificuldades de resolução do problema da PNL podem ser antevistos. Para o caso do sistema de KKT ser de posto deficiente, forma-se um problema ampliado, o qual é resolvido por mínimos quadrados, procedimento, este, inédito.
- *a direção de busca pode ser alterada quanto ao sentido de direção durante o procedimento de minimização da função de mérito:* esta, também, é uma abordagem nova, embora recentemente outros algoritmos (Lucia et ali, 1996) façam uso deste procedimento.
- *criação de parâmetros de sintonia:* o algoritmo possui dois parâmetros de sintonia. Um é com relação à estimativa inicial dos multiplicadores de Lagrange e o outro com relação aos valores que são impostos aos multiplicadores de Lagrange em determinadas situações, como problemas da PQ inviáveis, críticos e ardilosos.

Estabelecido o novo algoritmo denominado MISQPSOL, nos propusemos a analisá-lo. Ao iniciarmos a análise, verificamos a necessidade de melhor compreender os problemas da PQ não convexos, uma vez que esta é uma das linhas divisoras do nosso algoritmo em relação à maioria de algoritmos SQP existentes na literatura. Verificamos que poucos resultados encontram-se à disposição na literatura, aliás o que pode ser depreendido das afirmações, muitas vezes meramente qualitativas, em Lucia et ali (1996). Assim, procedemos a uma classificação das soluções de Karush-Kuhn-Tucker (KKT) estacionárias de problemas da PQ. A partir destas, pode-se prever o comportamento do algoritmo SQP e verificar se este pode falhar ou não. Esta contribuição, ao nosso ver, é grande. Ainda mais importante, cremos que é a utilização destes para a compreensão do comportamento dos algoritmos SQP. Incluem-se aqui aqueles que tratam de problemas da PQ convexos e não

convexos. Por exemplo, pode-se ter em mente porque é de interesse que se trabalhe com Hessianos indefinidos. Ou quando alterar o valor da direção de busca não traz vantagens.

Com relação à análise do algoritmo proposto, cremos que esta é uma contribuição menos significativa frente às outras, pois este é um assunto da alçada dos matemáticos, os quais, diga-se de passagem, efetuam análises bem mais abrangentes e rigorosas que as ora apresentadas. A importância deste assunto se deu na melhor compreensão do problema e por ser a fonte motivadora da análise da classificação das soluções da PQ, esta sim, sendo de grande relevância para o estabelecimento dos pontos dos algoritmos SQP que merecem destaque.

Com relação ao desempenho do algoritmo proposto, observamos que escolhidos os parâmetros de sintonia do algoritmo para um dado problema, a convergência do algoritmo para uma solução estacionária e viável dos problemas da PNL testados se deu em 100% dos casos. Estes problemas incluíram alguns bastante complexos como os exemplos de Psiaki & Park (1995) que apresentam um problema com acentuada curvatura e um outro com uma característica não convexa fortemente acentuada. Outros problemas sanados correspondem a gradientes de restrições nulos ou linearmente dependentes. Ainda, propusemos uma forma alternativa de classificar os problemas quanto à sua dificuldade inerente. Esta se mostrou mais eficiente em se prever dificuldades na solução do problema em relação às abordagens mais frequentemente usadas na literatura que são dependentes da dimensão relativa do problema da PNL. Basicamente, a idéia proposta é a de se analisar as funções existentes no problema da PNL e verificar que características estas têm. A cada uma destas características atribuímos um valor e a soma destas características define a complexidade do problema. Com isto, conseguimos definir adequadamente todos os problemas a exceção de um deles, o qual não apresenta nenhuma dificuldade aparente, a não ser a sua elevada não convexidade. Esta porém pode ser difícil de ser medida a priori e está relacionada à existência de vários problemas da PQ que definimos como críticos ou ardilosos.

Com relação à propositura da estratégia de otimização em tempo real, as inovações se devem mais à análise efetuada dos procedimentos adotados. Mostramos como a estratégia de otimização em uma e duas camadas pode ser realizada. Os procedimentos usados ainda

não constam de nenhuma publicação na literatura, podendo ser considerados originais. Contudo, esta não é uma característica importante, uma vez que qualquer engenheiro criativo poderia propor formas alternativas. O que quisemos ressaltar são as dificuldades que se encontram quando da sintonia da estratégia de otimização e a escolha das variáveis mais relevantes, incluindo a configuração de controle usada. Mostramos que a dinâmica do processo influencia o problema de otimização e notadamente a estratégia em uma camada pelo efeito da retroalimentação das variáveis controladas. Assim, a análise de robustez da estratégia final deve levar em conta o efeito da dinâmica do processo. Este é um aspecto que ao nosso ver, ainda, não foi referenciado em nenhum artigo e que merece atenção futura.

IV. A tese - forma de apresentação

A presente tese é apresentada, além da presente introdução, em mais sete capítulos e dois apêndices, os quais passam a ser descritos em seguida.

capítulo 2: **O problema da PNL e seus algoritmos de solução** - neste capítulo temos por objetivo efetuar uma revisão bibliográfica crítica do problema de programação não linear (PNL). Este é definido e as dificuldades de resolução são apontadas. Os algoritmos existentes são analisados e problemas para os quais eles podem falhar são apontados. A importância deste capítulo é em mostrar o porquê do objetivo principal da presente tese, a saber, o estabelecimento de um novo algoritmo de programação quadrática sucessiva.

capítulo 3: **O algoritmo MISQPSOL** - neste capítulo apresentamos com pormenores o algoritmo elaborado.

capítulo 4: **Análise do algoritmo MISQPSOL** - neste capítulo, objetivamos estudar o comportamento do algoritmo desenvolvido. Inicialmente, o problema da programação quadrática é caracterizado. Ainda, alguns resultados da teoria de sistemas lineares são apresentados. Com isto criam-se ferramentas que possibilitam caracterizar o funcionamento do algoritmo. A análise que é feita é com relação à convergência do algoritmo de solução da PQ para uma solução estacionária ou não. A partir destes resultados pode-se saber se a direção de busca gerada é adequada. Comentários são feitos com relação aos casos de falha do algoritmo.

capítulo 5: **Validação do algoritmo MISQPSOL**- neste capítulo usamos o algoritmo desenvolvido para resolver diversos problemas da literatura. Estes foram selecionados de forma a testar o algoritmo para diversas situações em que os algoritmos existentes na literatura podem não convergir para um solução estacionária ou até mesmo viável. Os problemas são de diferente natureza, sendo muitos deles da área de engenharia química. Alguns poucos são problemas meramente matemáticos, construídos com a finalidade de testar particularidades dos problemas da PNL. Alguns destes, como os exemplos 11 e 12 são de difícil solução. Selecionamos, ainda, diversas estimativas

iniciais e os parâmetros de sintonia do algoritmo foram variados. Com isto, conseguimos verificar a influência e dependência destes na resolução dos problemas. Os exemplos escolhidos são todos de programação não convexa e apresentam ordem de dificuldade variada. Propusemos uma forma alternativa à da literatura para prever o grau de dificuldade esperado para que se encontre a solução do problema. A medida foi capaz de prever diversas dificuldades, restando apenas um problema para o qual, a priori, não se tem medida de sua complexidade (exemplo 12). O algoritmo funcionou bem mostrando um desempenho quanto à robustez superior a de diversos algoritmos da literatura.

capítulo 6: **Otimização de um conversor FCC em tempo real - abordagem de uma e duas camadas** - neste capítulo mostramos como o problema de otimização em tempo real pode ser resolvido mediante o uso do algoritmo MISQPSOL. Propusemos algumas estratégias, três correspondentes à otimização em duas camadas e outras duas em uma camada, todas elas se mostrando factíveis. Para cada caso, mostramos como os modelos podem ser estabelecidos e também como as ações de controle podem ser calculadas. Mostramos que o problema da otimização em tempo real é bastante complexo e um desempenho adequado da estratégia proposta depende não só de uma sintonia adequada dos controladores usados, mas também da própria elaboração da estratégia. Por exemplo, a seleção de variáveis adequadas como variáveis controladas, ou, a não inclusão de determinadas variáveis, aparentemente não importantes para o problema econômico, fazem com que a estratégia proposta possa ter um desempenho não adequado. A existência de um algoritmo robusto para a resolução do problema da PNL mostrou ser uma ferramenta importante além de tornar o procedimento de estudo da otimização em tempo real bastante flexível. Ainda, mostramos que para o problema real diversos aspectos de sintonia devem ser levados em conta. Ainda, a solução ótima obtida para a estratégia em uma camada é fortemente dependente da sintonia. Ou seja, a retroalimentação, i.e., a própria dinâmica do sistema, provoca uma alteração do ponto de operação ótimo da planta. Esta é uma característica interessante que não é discutida na literatura e assim neste capítulo muitos aspectos de ordem prática são considerados.

Com isso, objetivamos mostrar os cuidados que devem ser tomados no procedimento da implementação de uma dada estratégia de otimização.

capítulo 7: **Conclusões e sugestões para continuidade da pesquisa** - neste capítulo mostramos quais as linhas de pesquisa iniciadas na presente tese. Realizamos um panorama do que foi feito e dos assuntos que devem ser estudados. Basicamente, a exposição mostra que a continuidade do presente trabalho pode se dar em duas linhas. Uma relativa à elaboração do algoritmo e a outra em relação ao estudo da estratégia de otimização. Com relação ao algoritmo, estudos são interessantes com relação a formas mais eficientes de se calcular a decomposição de matrizes, cálculo do Hessiano e atualização de matrizes. Ainda, relaxamentos no conjunto ativo ótimo podem ser desejados. Com relação a um aumento da robustez, é interessante que melhor se entenda o papel da função de mérito e que se utilize o fator de atenuamento para contornar determinados problemas de não convexidade acentuada. Estes aspectos não são triviais e merecem um estudo mais pormenorizado. Além disso, a fase de pós-processamento poderia ser mais elaborada ou até mesmo em algumas situações poder-se-ia forçar a matriz Hessiana a ser positiva definida no plano tangente às restrições ativas. Com relação à estratégia de otimização em tempo real, os estudos mais relevantes seriam com relação à robustez da estratégia final proposta. Ou seja, como a dinâmica do processo influencia a obtenção da solução ótima do problema. Além disso, a incorporação de distúrbios na planta é importante para que se tenha um melhor entendimento do real funcionamento das estratégias em otimização em uma e duas camadas.

capítulo 8: **Referências bibliográficas** - todos os artigos, teses e livros usados para a compreensão dos assuntos concernentes à tese ou referenciados são listados.

apêndice 1: **Apresentação das provas dos lemas do capítulo 4** - apresentamos as provas neste apêndice para facilitar a leitura do capítulo 4.

apêndice 2: **Exemplos usados na validação do algoritmo MISQPSOL** - os exemplos usados no capítulo 5 são listados e comentados.

CAPÍTULO 2: O PROBLEMA DA PNL E SEUS ALGORITMOS DE SOLUÇÃO

I. O problema da programação não linear (PPNL)

Define-se um problema de programação não linear (PPNL) como qualquer um que possa ser descrito por (P1).

encontre $x^* \in IR^n$ se este existe tal que:

$$f(x^*) = \min_{x \in X} f(x) \quad X = \{x | x \in X^o; h(x) = 0; g(x) \leq 0\} \quad (P1)$$

sendo, $f: IR^n \rightarrow IR; h: IR^n \rightarrow IR^m; g: IR^n \rightarrow IR^p$ e X^o é um espaço aberto contido em IR^n , f é denominada função objetivo e X é o espaço das restrições do PPNL.

Para o caso em que X e f são convexos, tem-se que o conjunto de soluções de (P1) é convexo (Mangasarian, 1969), ou seja, para quaisquer duas ou mais soluções x_i^* de (P1) o valor da função objetivo será o mesmo. Se ainda f for estritamente convexa então pode-se mostrar (Mangasarian, 1969) que a solução ótima de (P1) será única. No caso em que (P1) não é um problema de programação convexa tais propriedades já não se verificam e assim podem existir diversas soluções ótimas locais, isto é, a função objetivo assume um valor mínimo perto de uma certa vizinhança. As soluções ótimas locais podem ser obtidas a partir da resolução do problema (P2) dado a seguir:

encontre x^* se este existe tal que:

$$x \in B_\delta(x^*) \cap X \Rightarrow f(x) \geq f(x^*) \quad (P2)$$

sendo, $\delta > 0$ e $B_\delta(x^*)$ uma bola aberta em torno de x^* .

A resolução de (P1) ou (P2) é feita a partir de propriedades que devem ser satisfeitas pela solução do problema. O estabelecimento destas, as quais são denominadas de condições de otimalidade, depende de hipóteses e ainda hoje é alvo de pesquisa como pode ser visto, e.g., em Lee & Yung (1996), Blot & Michel (1996), Yezza (1996), Dontchev et ali (1995), Ye (1995), Cao (1995), Tapia & Tronet (1994), Zheng (1994), Ioffe (1989).

A resolução de (P1) é bastante mais complexa que a de (P2) e, assim, um procedimento freqüentemente adotado é resolver (P2). Neste texto, sempre que nos referirmos ao PPNL, estamos nos reportando a (P2). Analisando (P2), percebemos que nesta formulação está implícito que é possível descrever os pontos viáveis pertencentes à vizinhança viável definida por $B_\delta(x^*) \cap X$. Assim, seria interessante discutir como estes podem ser caracterizados. Se um ponto x pertence à vizinhança viável de x^* , então isto quer dizer que podemos andar ao longo de um caminho não linear partindo de x^* até atingir x . Este caminho é denominado de arco viável. Podemos melhor caracterizar o arco viável como sendo uma curva diferenciável parametrizada por um escalar v , a saber, $x = \zeta(v)$, obtida a partir de um ponto viável x^V qualquer de forma que:

$$\begin{aligned} \zeta(0) &= x^V \\ h(\zeta(v)) &= 0 \\ g(\zeta(v)) &\leq 0 \\ \frac{d\zeta}{dv}(0) &\neq 0 \\ \text{para todo } v: 0 &\leq v \leq \bar{v}; \bar{v} > 0 \end{aligned} \tag{2.I.1}$$

Percebemos que necessitamos estabelecer condições de existência de um arco viável. Estas deveriam ser preferencialmente obtidas a partir de propriedades conhecidas das funções que descrevem o problema (P2), e.g. os mapeamentos h, g . Às condições que os mapeamentos h, g devem satisfazer para que consigamos caracterizar os arcos viáveis chamamos de qualificações de restrições. Em Nemhauser et ali (1989), uma qualificação de restrições é mostrada para o caso em que as restrições são diferenciáveis em X , a qual é apresentada a seguir:

“A condição de qualificação é válida em x^V se para qualquer vetor não nulo d com $\nabla h^T(x^V)d = 0$; $\nabla g^T(x^V)d \leq 0$, d é tangente a um arco viável duplamente diferenciável originado em x^V .”

Uma qualificação de restrições bastante usada é a de se ter os componentes dos gradientes das restrições linearmente independentes (LI). O enunciado desta qualificação é dado em (C1) e faz uso da definição de ponto regular.

definição 2.I.1: *do ponto regular* - um ponto x é dito regular quanto às restrições definidas pelos mapeamentos h e g se $h(x) = 0$; $g(x) \leq 0$ e se os componentes dos gradientes de h e g avaliados em x são LI.

(C1) x^* é ponto regular das restrições $h(x)=0$ e $g(x)\leq 0$.

Outras condições de qualificação de restrições podem ser encontradas e.g. em Mangasarian (1969) ou Bazaraa et al (1993). Em seguida, apresentamos duas condições de otimalidade extraídas de Mangasarian (1969). O teorema 2.I.2 é particularmente importante e a condição (KKT) é a mais freqüentemente empregada pelos algoritmos de solução do PPNL.

teorema 2.I.1: (condição de otimalidade suficiente do problema (P1)) - Seja $L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)$ com $\lambda \in \mathbb{R}^m$; $\mu \in \mathbb{R}^p$. Se existem $x^* \in X^o$; $\lambda^* \in \mathbb{R}^m$; $\mu^* \in \mathbb{R}^p$; $\mu^* \geq 0$ tais que a condição C2 se verifica para todos $\lambda \in \mathbb{R}^m$; $x \in X^o$ e $\mu \geq 0$; $\mu \in \mathbb{R}^p$ então x^* é solução de (P1).

$$(C2) \quad L(x^*, \lambda, \mu) \leq L(x^*, \lambda^*, \mu^*) \leq L(x, \lambda^*, \mu^*)$$

teorema 2.I.2: (condição de otimalidade necessária para casos particulares dos problemas (P1) ou (P2)) - Seja x^* uma solução global de (P1) ou uma solução local de (P2). Sejam f , h , g diferenciáveis em x^* . Se h ou g

satisfazem (C3) ou (C4) ou (C5) então existem $\lambda^* \in \mathbb{R}^m$ e $\mu^* \in \mathbb{R}^p$ tais que a condição (KKT) se verifica.

$$(KKT) \quad : \quad \begin{cases} \nabla f^T(x^*) + \lambda^{*T} \nabla h^T(x^*) + \mu^{*T} \nabla g^T(x^*) = 0 \\ g(x^*) \leq 0 \\ h(x^*) = 0 \\ \mu^{*T} g(x^*) = 0 \\ \mu^* \geq 0 \end{cases}$$

(C3) qualificação de Karush-Kuhn-Tucker em x^* :

$$\left. \begin{array}{l} y \in \mathbb{R}^n \\ \nabla g_I^T(x^*) y \leq 0 \\ \nabla h^T(x^*) y = 0 \\ I = \{i | g_i(x^*) = 0\} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \exists e: E \rightarrow \mathbb{R}^n; E = [0,1] \subset \mathbb{R}; \text{ t.q.} \\ e(0) = x^* \\ e(\tau) \in X; \tau \in E \\ e \text{ e' diferenciavel em } \tau = 0 \text{ e:} \\ \frac{de(0)}{d\tau} = v y \text{ para algum } v > 0 \end{array} \right.$$

(C4) qualificação fraca de Arrow- Hurwicz-Uzawa:

- h é tanto pseudoconvexa como pseudocôncava em x^*
- $\left\langle \begin{array}{l} \nabla g_Q^T(x^*) z > 0 \\ \nabla g_P^T(x^*) z \geq 0 \\ \nabla h^T(x^*) z = 0 \end{array} \right\rangle$ tem uma solução $z \in \mathbb{R}^n$, sendo P e Q

definidos por:

$$P = \{i | g_i(x^*) = 0; g_i \text{ e' pseudocôncava em } x^*\}$$

$$Q = \{i | g_i(x^*) = 0; g_i \text{ não e' pseudocôncava em } x^*\}$$

(C5) qualificação de Arrow-Hurwicz-Uzawa modificada:

- h é continuamente diferenciável em x^*
- os componentes de $\nabla h(x^*)$ são linearmente independentes

- o sistema $\left\langle \begin{array}{l} \nabla g_I^T(x^*) z > 0 \\ \nabla h^T(x^*) z = 0 \end{array} \right\rangle$ tem solução $z \in \mathbb{R}^n$.
- $$I = \{i | g_i(x^*) = 0\}$$

Ou seja, se um dado ponto descrito por $\{x, \lambda, \mu\}$ satisfaz (KKT) então ele é um forte candidato a ser solução de (P2). Note, porém, que o teorema diz que para que o ponto $\{x, \lambda, \mu\}$ seja solução de (P2), ele deve satisfazer (KKT). Ou seja, se obtemos um ponto que

satisfaça (KKT) nada garante que ele seja de fato a solução procurada. Para se ter certeza de que um ponto que satisfaça (KKT) e as condições de qualificação seja solução de (P2), outras condições devem ser estabelecidas, as chamadas condições de otimalidade suficientes.

Resta, pois, tratarmos das condições suficientes de otimalidade. Para tanto iremos fazer uso da definição de plano tangente, denotado por Π , dada a seguir.

definição 2.I.2: *do plano tangente Π* - Sejam as restrições ativas no ponto x denotadas por

$q(x)=0$, $q(x) = \{h(x); g_I(x)\}$; $I = \{i | g_i(x) = 0\}$. Sejam, ainda, h e g diferenciáveis em X^0 . O plano tangente Π às restrições $q(x)=0$ é definido por:

$$\Pi = \{y: \nabla q(x)y = 0; y \in \mathbb{R}^n \text{ e } y \neq 0\}.$$

O seguinte teorema é extraído de Mangasarian (1969).

teorema 2.I.3: (condição suficiente de otimalidade) Sejam X^0 um espaço aberto, f, h, g definidas como em (P2). Sejam, ainda, $x^* \in X^0$ e o conjunto I dado por $I = \{i | g_i(x^*) = 0\}$. Sejam ainda f, h, g_I respectivamente pseudoconvexa, diferenciável pseudoconvexa e pseudocôncava, diferenciável pseudoconvexa em x^* . Se existem $\lambda^*, \mu^* \in \mathbb{R}^n$ tais que $\{x^*, \lambda^*, \mu^*\}$ satisfazem (KKT) então x^* é solução de (P2).

Outros enunciados do teorema 2.I.3 podem ser feitos, a saber:

variante 1: substituir as condições sobre f, h, g_I pela seguinte: para $y \in \Pi \Rightarrow y^T \nabla^2 L(x^*, \lambda^*, \mu^*) y > 0$, condição esta comumente expressa como o Hessiano do Lagrangeano na solução ótima deve ser positivo definido (PD) no plano tangente.

variante 2: substituir a condição anterior por $\bar{Z}^T \nabla^2 L(x^*, \lambda^*, \mu^*) \bar{Z}$ deve ser PD, sendo que \bar{Z} é uma base para o kernel das restrições ativas na solução ótima.

As condições acima são suficientes, i.e., quando elas se verificam, caracteriza-se uma solução de (P2). São frequentemente chamadas de condições suficientes de segunda ordem. Uma condição mais fraca é a de que a característica de serem as matrizes nas variantes 1 e 2 PD seja substituída pela propriedade de serem elas positivas semidefinidas (PSD), tendo-se, então, a condição necessária de segunda ordem.

II. Algoritmos de solução do PPNL

Na presente exposição, como é de praxe, chamaremos de algoritmo a uma dada maneira de se aplicar um método de resolução do problema (P1) ou (P2) e código à implementação computacional do algoritmo proposto.

Lootsma (1985) ressalta que um bom algoritmo para a resolução do problema (P1) ou (P2) deve ter as seguintes características desejáveis:

1. o algoritmo deve ser robusto, i.e., deve ser capaz de resolver o maior número de problemas da PNL
2. o algoritmo deve ser capaz de lidar com problemas pequenos, médios e grandes
3. o algoritmo deve ser eficiente

Assim, a escolha de um algoritmo deve ser precedida da escolha de um dado método de resolução. Faz-se necessário, então, conhecer as diferentes maneiras de se resolver o problema da PNL, para que então um algoritmo possa ser escolhido.

De uma forma geral, todos os métodos numéricos de resolução do problema (P1) ou (P2) fazem uso de um esquema iterativo em que a partir de uma estimativa inicial qualquer (x^o, λ^o, μ^o) obtém-se uma seqüência de valores $\{x_k, \lambda_k, \mu_k\}$ que converge para a solução ótima (x^*, λ^*, μ^*) do problema (P1) ou (P2). O que distingue um método do outro é a forma com que esta seqüência de valores é calculada.

Os problemas reais são em geral de programação não convexa, ou seja, podem existir diversas soluções ótimas locais, i.e., soluções do problema (P2). Assim, os algoritmos que se baseiam nas condições de (KKT) resolvem o problema (P2) ao invés do (P1). A obtenção da solução (global) de (P1) é bem mais complexa e não será alvo de estudo da presente tese.

Os algoritmos de otimização normalmente transformam o problema em subproblemas solúveis. Os primeiros métodos que surgiram na literatura (método das penalidades, de barreiras e da função Lagrangeana aumentada) objetivavam transformar o problema

original em um problema irrestrito. Isto, porque métodos para a solução do problema irrestrito, eram conhecidos.

A idéia por trás dos métodos que resolvem problemas irrestritos é gerar uma seqüência de valores $\{x_k\}$ como:

$$x_{k+1} = x_k + \alpha_k d_k$$

onde, d_k indica uma direção de busca descendente e α_k a amplitude que se irá deslocar nesta direção. Se o gradiente da função objetivo ($\nabla f(x)$) for não nulo em x_k então d_k e α_k podem ser escolhidos de modo que $f(x_{k+1}) < f(x_k)$ (Gill et ali, 1988). A incorporação de restrições lineares e de limites nas variáveis não afeta este procedimento descrito. Este grupo de métodos recebe a denominação de métodos descendentes.

Uma alternativa a estes métodos são os métodos dos gradientes reduzidos e projeção do gradiente, para os quais o problema irrestrito é formado reduzindo-se as variáveis em função das restrições. Segundo Gill et ali (1988) este grupo de métodos não se mostra eficiente para problemas com restrições fortemente não lineares, ao contrário do que dito em Lasdon & Waren (1983). Um outro inconveniente destes métodos é a sua difícil implementação.

Posteriormente, desenvolveram-se outros métodos (método da projeção do Lagrangeano, algoritmo SQP), os quais transformam o problema original restrito em subproblemas restritos mais simples. Ainda, uma outra forma de se distinguir os diferentes algoritmos é quanto ao uso da função Lagrangeana. Alguns (método das penalidades, de barreiras e método generalizado dos gradientes reduzidos) usam a função Lagrangeana implicitamente, enquanto outros trabalham diretamente com esta (método da projeção do Lagrangeano, função Lagrangeana aumentada, algoritmos de programação quadrática sucessiva (SQP)), (Gill et ali, 1988). Um outro grupo de métodos que também é citado para a resolução da PNL é o algoritmo SLP (programação linear sucessiva). A idéia básica deste é linearizar-se o PNL em torno de um ponto e aplicar a programação linear (PL) para a sua resolução. Na nova iteração, efetua-se uma nova linearização em torno do último ponto

obtido e prossegue-se com o algoritmo. A vantagem deste é a sua fácil implementação. Quando a solução ótima encontra-se no vértice, a convergência é rápida. Além disso, problemas grandes podem ser tratados. Entre as desvantagens deste, citam-se a lenta convergência para problemas em que a solução ótima não se encontra nos vértices e o fato de que as restrições não lineares são violadas enquanto se estiver longe da solução ótima (Lasdon & Waren, 1983).

Descrições sobre o funcionamento destes métodos podem ser encontradas em Gill et ali (1988), Gill et ali em Nemhauser (1989), Papalambros & Wilde (1988) ou Bazaraa et ali (1993).

Percebemos que inúmeros métodos existem. Para problemas pequenos e de médio porte o método SQP é considerado o melhor dentre eles por inúmeros autores (Gill et ali 1988, Conn et ali, 1991, Heinz & Spellucci, 1994). Para problemas de grande porte não há consenso sobre o melhor método a ser empregado e esforços vem sendo feitos para melhorar o desempenho dos algoritmos SQP e também no desenvolvimento de algoritmos pertencentes ao método do Lagrangeano aumentado (Schmid & Biegler, 1994, Conn et ali, 1991, Bartholomew-Biggs & Hernandez, 1995, Psiaki & Park, 1995, Lucia et ali, 1996). Como os problemas a serem resolvidos na tese são de médio e pequeno porte, optamos pela escolha do método SQP. Ainda, um algoritmo definitivo não existe. Problemas de convergência podem existir e inúmeras adaptações visando-se a um aumento de eficiência podem ser feitas. Em função disto partimos para o estabelecimento de um novo algoritmo SQP.

III. Os algoritmos SQP

III.1 Introdução

Credita-se a origem dos algoritmos de programação quadrática sucessiva (SQP) a Wilson (1963), tendo sido, no entanto, estendidos e generalizados apenas nos anos de 1972 a 1977. Desde então, é notória a posição de destaque por eles ocupada entre os algoritmos de resolução do problema da programação não linear (PNL). Isto se deve ao êxito destes algoritmos na resolução de problemas da PNL de pequeno e médio porte.

Muitos anos se passaram e ainda não existe um algoritmo SQP definitivo, uma vez que o problema que se deseja resolver é complexo. Embora a concepção do método SQP seja conceitualmente simples, a implementação do algoritmo requer cuidados. Cuidados, estes, necessários para que a complexidade inerente aos problemas da PNL (PPNL) não ocasione dificuldades tamanhas que impossibilitem a sua resolução.

Refletindo sobre os aspectos destacados nos parágrafos precedentes percebemos que a compreensão do método requer que se entenda a idéia do algoritmo e como ela é concretizada. Ou seja, urge que se defina a classe de algoritmos SQP e como estes resolvem o problema da PNL.

Como qualquer um dos outros algoritmos de resolução de PPNL, os algoritmos SQP correspondem a um método iterativo que gera uma seqüência de valores $\{x_k, \lambda_k, \mu_k\}$ que converge para a solução do problema (P2) dada por (x^*, λ^*, μ^*) . A diferença para os demais métodos reside, pois, em como esta seqüência de valores é gerada.

O estabelecimento desta seqüência pode ser interpretado de formas diferentes. Duas delas merecem atenção especial. Objetiva-se obter a solução (x^*, λ^*, μ^*) de (P2) e sabe-se que esta deve satisfazer as condições de KKT. Por outro lado, as condições de KKT nada mais são que um sistema de equações não lineares. Assim, obter a solução ótima de (P2) nada mais é que resolver um sistema de equações $F(x, \lambda, \mu) = 0$. Desta forma uma primeira interpretação surge em Stoer (1985), a saber, os métodos SQP são essencialmente métodos

de Newton ou quasi-Newton de resolução de $F(x, \lambda, \mu) = 0$ dado pelas condições de KKT. Resta pois analisarmos como este sistema de equações é resolvido.

Wright (1989) define os métodos SQP como sendo aqueles em que para cada iteração do algoritmo efetua-se uma aproximação linear do conjunto X definida como $X(x_k)$. Uma função quadrática é então minimizada sobre este conjunto simplificado, obtendo-se os novos valores da sequência $\{x_k, \lambda_k, \mu_k\}$. Esta minimização de uma função quadrática sujeita a restrições lineares corresponde a um problema de programação quadrática (PQ). Assim, devemos analisar como este deve ser formulado. Palomares & Mangasarian (1976) argumentam que se construirmos o problema da (PQ) de forma que a função Lagrangeana deste seja igual a uma aproximação quadrática do Lagrangeano do problema (P2), então a solução da (PQ) é a solução de (KKT) de uma aproximação quadrática do Lagrangeano de (P2) na iteração considerada. Se o termo quadrático da função objetivo for igual ao Hessiano do Lagrangeano então tem-se o método de Newton, caso o primeiro seja apenas uma aproximação do último tem-se um método quasi-Newton.

Em função do que foi exposto temos que a classe de algoritmos SQP, é caracterizada pela geração da sequência $\{x_k, \lambda_k, \mu_k\}$ como segue:

1. Resolva o seguinte problema da PQ para obter d_k :

$$\begin{aligned} \min \frac{1}{2} d_k^T H(x_k, \lambda_k, \mu_k) d_k + \nabla f^T(x_k) d_k \\ \text{s.a. } \nabla h^T(x_k) d_k = -h(x_k) \\ \nabla g^T(x_k) d_k \leq -g(x_k) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min \frac{1}{2} d_k^T H(x_k, \lambda_k, \mu_k) d_k + \nabla f^T(x_k) d_k \\ \text{s.a. } x_k + d_k \in X(x_k) \end{aligned}$$

onde, H é uma aproximação do Hessiano do Lagrangeano de (P1) ou (P2) avaliado na iteração k . $X(x_k)$ é a aproximação linear de X na iteração k , obtida expandindo-se as restrições em série de Taylor e truncando-se os termos de ordem superior.

2. $x_{k+1} = x_k + d_k$
3. λ_{k+1} e μ_{k+1} correspondem aos multiplicadores de Lagrange do problema da PQ.

Percebemos que a essência do método SQP é simples. No entanto, sendo um método de Newton a convergência para a solução do problema (P2) será local sob certas hipóteses, i.e., a sequência $\{x_k, \lambda_k, \mu_k\} \rightarrow (x^*, \lambda^*, \mu^*)$ para valores (x_k, λ_k, μ_k) suficientemente próximos da solução ótima. Como não se conhece a solução ótima a priori é necessário que se efetuem mudanças no algoritmo básico de forma a que se tenha convergência global para uma solução de KKT, independentemente do ponto de partida. Se o espaço definido por X for convexo então se a sequência for construída como: $x_{k+1} = x_k + \alpha_k d_k \in X$ para todo $\alpha_k \in [0,1] \subset \mathbb{R}$, temos que x_{k+1} pertencerá a X (Wright, 1989). Pode-se, então, definir um problema de minimização em linha de forma a se garantir a propriedade de convergência global do método SQP para esta situação (Wright, 1989). Este é o ponto de partida do método SQP, ou seja, introduz-se uma penalidade no passo 2, o qual passa a ser definido como:

$$x_{k+1} = x_k + \alpha_k d_k$$

sendo, α_k um amortecimento obtido a partir da minimização unidirecional de uma função objetivo adequada, denominada função de mérito.

Percebemos que diversas particularidades implícitas na exposição existem e delas depende o bom funcionamento do algoritmo, a saber:

1. É necessário se calcular os gradientes dos mapeamentos f, h, g .
2. É necessário se obter uma boa aproximação do Hessiano da função Lagrangeana.
3. É necessário se resolver o problema da programação quadrática e estabelecer se este terá solução.
4. É necessário se definir a função de mérito e resolver um problema de busca unidirecional.
5. É necessário se estabelecer um critério de parada, i.e., quando o algoritmo convergiu para a solução ótima.

Para que melhor se entendam as particularidades apresentadas e a partir destas as diferenças entre os diversos algoritmos SQP, cada um dos tópicos será, em seguida, abordado.

III.2 Cálculo dos gradientes dos mapeamentos f, h, g

Para o método SQP é vital que se calculem gradientes dos mapeamentos f , h , g em cada iteração em x_k , a saber, $\nabla f(x_k), \nabla h(x_k), \nabla g(x_k)$.

Citam-se quatro métodos principais para o cálculo dos gradientes (Averick et ali, 1994) no método SQP: uso de funções analíticas, uso do método de diferenças finitas, uso de manipuladores simbólicos e uso da diferenciação automática. Destes o mais comumente empregado é o das diferenças finitas, embora o primeiro e o último sejam mais precisos. O método de diferenças finitas se divide em dois grupos principais, as abordagens central e dianteira, sendo a primeira a mais amplamente usada. Pormenores de como a implementação computacional deve ser feita podem ser encontrados em Nemhauser (1989) ou Papalambros & Wilde (1988).

III.3 Cálculo do Hessiano

O cálculo do Hessiano é uma etapa fundamental do algoritmo SQP e dela dependerá fortemente a convergência do algoritmo. Por exemplo se H_k coincide com a expressão analítica, então o algoritmo SQP poderá ter convergência local quadrática (Palomares & Mangasarian, 1976). Se H_k é calculado por diferenças finitas, o algoritmo pode ter convergência local R-superlinear (Palomares & Mangasarian, 1976) ou quadrática (Bazaraa et ali, 1993) dependendo das hipóteses feitas. Comportamento semelhante deve ser esperado se é feito uso do método da diferenciação automática para avaliar H_k . Contudo estes métodos podem apresentar alguns inconvenientes. Durante uma iteração qualquer H_k pode ser PSD, ND, NSD, indefinido ou até mesmo uma matriz nula. Neste caso, problemas durante a resolução da PQ podem surgir quando H_k sobre o plano tangente às restrições ativas não for PD ou pelo menos PSD. Nestes casos, pode não existir uma solução finita para o problema da PQ, ou ainda, que se terá infinitas soluções locais. Assim, uma propriedade interessante que H_k deveria apresentar é a de que esta fosse PD. Em função desta característica interessante, partiu-se para o desenvolvimento de métodos que atualizassem H_k mantendo a condição desta ser PD. Coleman (1984) efetua uma excelente discussão de como estas atualizações podem ser obtidas. Refinamentos podem ser encontradas em outras referências bem como a relação das expressões obtidas com a convergência do algoritmo (Stoer, 1985, Palomares & Mangasarian, 1976, Nocedal &

Overton, 1985). Em verdade, forçar o Hessiano a ser PD pode fazer com que se piore a taxa de convergência do algoritmo. Um exemplo disto pode ser visto no problema dado em (2.III.3.1) para o qual, a partir de uma certa iteração, pode-se mostrar que a direção de busca gerada pelo algoritmo SQP básico com o uso de expressões analíticas para o cálculo do Hessiano é dada por (2.III.3.2). Para este problema o Hessiano analítico para qualquer iteração do algoritmo será sempre NSD. Se forçarmos a matriz a ser PD, ou ainda, PSD, i.e., alterando apenas o segundo elemento de sua diagonal principal, teremos que os multiplicadores de Lagrange calculados pelo algoritmo SQP a partir de uma certa iteração serão negativos e irá se resolver durante um certo tempo um problema da PQ irrestrito e nestas condições a direção de busca será dada por (2.III.3.3), resultando em um valor menor que o obtido por (2.III.3.2). Ou seja, neste exemplo percebe-se claramente o deterioramento no desempenho do algoritmo.

$$\begin{aligned} \min x_2 \\ \text{s.a. } 1 + x_1 - x_2^2 &\leq 0 \\ 1 - x_1 - x_2^2 &\leq 0 \\ x_2 &\geq 1/2 \end{aligned} \tag{2.III.3.1}$$

$$d_{2,k} = \frac{1 - x_{2,k}^2}{2x_{2,k}} \tag{2.III.3.2}$$

$$d_{2,k} = -\frac{1}{h_{2,2,k}} \tag{2.III.3.3}$$

onde, $h_{2,2,k}$ é o segundo elemento da diagonal principal do Hessiano avaliado na iteração k .

Além disso, para determinados problemas forçar H_k a ser PD pode fazer com que o algoritmo SQP não convirja para uma solução de mínimo de (P2) (Lucia et ali 1996).

O cálculo do Hessiano pelo método das diferenças finitas pode se processar de diversas formas, sendo as mais corriqueiras as centrais, dianteiras e aquelas que utilizam informações dos gradientes. Esta última pode apresentar erros numéricos significativos quando os gradientes também são avaliados pelo método das diferenças finitas. Maiores pormenores sobre questões de implementação podem ser obtidas, e.g., em Nemhauser (1989).

Uma outra abordagem bastante popular é o uso dos métodos quasi-Newtonianos. Estes se baseiam em um procedimento em que a partir de uma estimativa do Hessiano obtida em um ponto x_k , diga-se, $\nabla^2 T(x_k)$, obtém-se uma nova estimativa em um ponto x_{k+1} , a saber, $\nabla^2 T(x_{k+1})$, resolvendo-se um problema do tipo (2.III.3.4).

$$\begin{aligned}
 & \min \|\nabla^2 T(x_{k+1}) - \nabla^2 T(x_k)\| \\
 & \text{s.a. } \nabla^2 T(x_{k+1})d_k = z_k \\
 & \quad z_k = \nabla T(x_{k+1}) - \nabla T(x_k) \\
 & \quad \nabla^2 T(x_{k+1}) = \nabla^2 T(x_{k+1})^T
 \end{aligned} \tag{2.III.3.4}$$

sendo, que $x_{k+1} = x_k + d_k$ e $\|\cdot\|$ é uma norma qualquer e em geral usa-se a norma de Frobenius. A idéia do problema é tentar minimizar a distância entre as matrizes de uma iteração à outra. Esta distância pode eventualmente ser avaliada pela inversa da matriz Hessiana, ou, então, ponderando-se a distância. Adicionalmente, pode-se incorporar restrições de que a matriz Hessiana deve permanecer PD ou manter uma determinada estrutura esparsa.

Dependendo de como a função objetivo em (2.III.3.4) é formulada, i.e., que norma é usada e como as matrizes são ponderadas diversos métodos para o cálculo do Hessiano existem, como os métodos PSB, DFP, BFGS. Dentre estes, o método BFGS é um dos mais populares e aparentemente apresenta um melhor desempenho em termos de tempo computacional requerido, quando em uso com o algoritmo SQP (Stoer, 1985). Basicamente a fórmula do método BFGS é apresentada em (2.III.3.5).

$$\nabla^2 T(x_{k+1}) = \nabla^2 T(x_k) + \frac{z_k z_k^T}{d_k^T z_k} - \frac{\nabla^2 T(x_k) d_k d_k^T \nabla^2 T(x_k)}{d_k^T \nabla^2 T(x_k) d_k} \tag{2.III.3.5}$$

Quando $\nabla^2 T(x_k)$ for PD e $d_k^T z_k > 0$ então a nova estimativa do Hessiano permanecerá PD. Um problema que surge é que pode não ser possível manter $d_k^T z_k > 0$ (Gill et ali em Nemhauser, 1989). Por causa disto inúmeras mudanças na formulação (2.III.3.5) podem ser encontradas na literatura.

As formulações básicas mais freqüentemente usadas são apresentadas de uma forma bastante completa em Coleman (1984) e Stoer (1985), os quais também fazem considerações com relação à introdução de restrições de esparsidade. Bartholomew-Biggs & Hernandez (1995) comentam algumas abordagens recentes para formulações que levam em conta a esparsidade de problemas de grande porte. Dependendo de como as matrizes são atualizadas diferentes taxas de convergência para o algoritmo SQP são obtidas. Palomares & Mangasarian (1976) estabelecem condições gerais para que uma dada aproximação faça com que o algoritmo SQP tenha taxas de convergência linear, superlinear ou quadrática. As condições se baseiam em erros relativos à matriz obtida por expressões analíticas. Han (1976) e Stoer (1985) apresentam taxas de convergência para diferentes fórmulas de atualização, incluindo a formulação BFGS.

Formas com que as atualizações podem ser implementadas podem ser vistas em Nocedal & Overton (1985), os quais também efetuam uma comparação em termos de eficiência computacional. Modificações sobre a fórmula BFGS de modo com que se minimizem problemas de indefinição e outras questões numéricas, podem ser encontradas em Powell & Yuan (1986), Lucia & Xu (1990) ou, ainda, em Heinz & Spellucci (1994). Outras considerações com relação aos métodos de cálculo de Hessianos podem ser encontradas e.g. em Nocedal (1990), Dennis & Schnabel em Nemhauser (1989). Referências adicionais para os casos esparsos e grandes podem ser encontradas e.g. em Averick et.al. (1994), Coleman (1984), e Dennis & Schnabel em Nemhauser (1989). Bartholomew-Biggs & Hernandez (1995) discutem o procedimento de forçar a matriz Hessiana a ser PD no plano tangente às restrições ativas. Uma idéia é a de se introduzir uma perturbação em alguns dos elementos da diagonal da matriz Hessiana de forma a que se transforme esta em uma matriz PD. Os autores referenciam um resultado de Gould (1986) que diz que se o número de autovalores negativos em (2.III.3.6) não for maior que o número de linhas correspondentes às restrições ativas, o Hessiano será positivo definido no plano tangente às restrições ativas. Assim, pode-se limitar o número de valores que serão alterados na matriz Hessiana. Ainda, os autores discutem como o procedimento de introduzir perturbações nos elementos da diagonal pode ser efetuado concomitantemente com a resolução do sistema de (KKT). Resultados são apresentados que mostram que é de grande interesse que se possa trabalhar com a matriz Hessiana indefinida.

$$\begin{bmatrix} H_k & A^T \\ A & 0 \end{bmatrix} \quad (2.III.3.6)$$

onde, A indica o conjunto de restrições do problema da PQ que estão ativas.

Lucia et ali (1996) mostram um procedimento alternativo mais simples. A verificação da condição de H_k é feita sobre a direção de busca obtida. Se $d_k^T H_k d_k < 0$, então efetua-se um tratamento especial de forma a se evitar que se obtenha uma direção de busca não adequada a nível da resolução da PQ.

III.4. O problema da PQ e sua solução

A programação quadrática é caracterizada de uma forma genérica pelo seguinte problema:

$$\min_{d \in X_I} \frac{1}{2} d^T H d + c^T d; X_I = \left\{ d \mid A_E d = b_E; A_I d \leq b_I; A_E \in \mathbb{R}^{m \times n}; A_I \in \mathbb{R}^{p \times n} \right\} \quad (2.III.4.1)$$

As condições de otimalidade expressas por (KKT) para este problema são dadas em (2.III.4.2a) e (2.III.4.2.b).

$$\begin{bmatrix} H & A_E^T & A_{I_j}^T \\ A_E & 0 & 0 \\ A_{I_j} & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \\ \mu_{I_j} \end{bmatrix} = \begin{bmatrix} -c \\ b_E \\ b_{I_j} \end{bmatrix}; I_j = \{i \mid A_i d = b_i\} \quad (2.III.4.2a)$$

$$\mu_{I_j} \geq 0 \quad (2.III.4.2b)$$

onde, A_i e b_i indicam a $i^{\text{ésima}}$ linha de A_I e b_i , respectivamente, I_j indica o conjunto de inequações que estão ativas, doravante denominado simplesmente por conjunto ativo.

Note que para as nossas finalidades, o interesse em se resolver (2.III.4.1) é obter uma direção de busca para o algoritmo SQP. Ou seja, (2.III.4.1) é obtido como um modelo linearizado do problema da PNL (P1) ou (P2). Uma primeira idéia que surge para se obter um modelo linearizado de (P1) ou (P2) é adotar H, A_E, A_I, c, b_E, b_I como em (III.4.3):

$$\begin{aligned}
 H &= H(x_k, \lambda_k, \mu_k) \\
 A_E &= \nabla h^T(x_k) \\
 A_I &= \nabla g^T(x_k) \\
 b_E &= -h(x_k) \\
 b_I &= -g(x_k)
 \end{aligned} \tag{2.III.4.3}$$

Uma vez resolvido (2.III.4.1) com as definições de (2.III.4.3), as novas estimativas das variáveis de (P1) ou (P2) podem ser obtidas como:

$$\begin{aligned}
 \lambda_{k+1} &= \lambda \\
 \mu_{k+1} &= \mu \\
 d_k &= d^*
 \end{aligned}$$

onde, o índice k se refere à $k^{\text{ésima}}$ iteração do algoritmo SQP e λ , μ e d^* são as soluções de (2.III.4.1).

A obtenção de um d^* que satisfaz (2.III.4.2) não implica em se ter chegado na solução de (2.III.4.1). Isto porque se H não for PD ou PSD no plano tangente às restrições ativas, a solução será estacionária mas não de mínimo. Assim, é interessante se caracterizar como as soluções obtidas em (2.III.4.2) podem ser classificadas. Isto será apresentado com bastante pormenores no capítulo 4. No momento, o nosso objetivo primordial é o de estabelecer como (2.III.4.2) pode ser resolvido numericamente e que tipos de problemas podem surgir para a obtenção de uma solução estacionária. Para efeito de discussão, assumiremos que a solução ótima d^* é obtida a partir de um processo iterativo, em que resolve-se (2.III.4.2a) para diferentes I_j até que (2.III.4.2b) seja satisfeito e este será o algoritmo de conjuntos ativos considerado.

Se X_I representa uma linearização das restrições de (P1) ou (P2) em um ponto viável x^V , i.e., $h(x^V) = 0$; $g(x^V) \leq 0$, então se caminharmos de x^V para um novo ponto, i.e., $x_k = x^V$ e $x_{k+1} = x_k + d^*$, nada garante que a linearização feita, continue válida para x_{k+1} , i.e, que x_{k+1} seja viável, ou que a nova linearização em x_{k+1} seja *consistente*, sendo que o termo *consistente* será melhor caracterizado ao longo da discussão que segue.

Quando da realização da linearização do problema da PNL, ainda que em torno de um ponto viável, diversos problemas surgem, os quais poderíamos classificar em dois grupos iniciais:

- **problema 1** - *existência de restrições redundantes, i.e., não necessárias ao problema da PQ*: a inclusão de restrições redundantes aumenta a dimensão do problema desnecessariamente. Além disso, dependendo de como o problema da PQ é resolvido pode-se não encontrar a solução do problema. É necessário, pois, analisar as restrições e se possível desconsiderar aquelas que sejam redundantes.
- **problema 2** - *possibilidade de se gerar um problema da PQ que não tenha solução viável, i.e., para os quais X_l é vazio*: se o problema da PQ é usado para gerar uma direção de busca do problema da PNL, então a inexistência de uma solução viável pode afetar o algoritmo SQP. Deve-se, pois, buscar um meio de se contornar esta situação.

Na figura 2.III.4.1, extraída de Boneh et ali (1993), apresentamos uma região viável descrita por restrições de desigualdade. A região viável é definida pelas restrições 1,4,3 e uma dentre as restrições 2 e 7. As restrições que, retiradas, alteram a região viável são ditas necessárias. Assim, no caso temos que as restrições 1, 4 e 3 são necessárias. Percebemos, assim, que a restrição 5 é irrelevante para a descrição da região viável. Neste caso, ela recebe o nome de restrição absolutamente fortemente redundante (ASR “absolutely strongly redundant”). Já a restrição 6, inobstante redundante, faceia a região viável. Suponha que a solução ótima do problema esteja no vértice A da figura 2.III.4.1 Neste caso, a restrição 6 ativada poderá não afetar o algoritmo de solução, i.e., se ela for ativada em conjunto com a restrição 1 ou 4, a solução ótima pode, eventualmente ser obtida. Este tipo de restrição recebe o nome de absolutamente fracamente redundante (AWR “absolutely weakly redundant”). Já as restrições 2 e 7, se forem ambas retiradas afetam a descrição da região viável. No entanto, apenas uma delas é necessária sendo, a outra redundante. Restrições deste tipo recebem o nome de relativamente fracamente redundantes (RWR “relatively weakly redundant”). Uma definição mais rigorosa e pormenorizada de restrições é apresentada em Boneh et ali (1993). Para os nossos propósitos o que nos interessa é saber que a solução ótima do problema da PQ poderá ser determinada por restrições do tipo necessárias, AWR e RWR. Considerando que o algoritmo de conjuntos ativos seja capaz de identificar restrições ativas linearmente dependentes (e conseqüentemente as restrições RWR serão detectadas), então, o que se faz

necessário é eliminar as restrições ASR e para aquelas do tipo AWR, verificar se a sua presença não afeta a solução do problema.

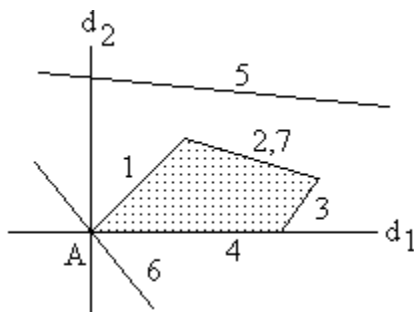


figura 2.III. 4.1: representação de um espaço típico viável da PQ

Retiradas todas as restrições redundantes, restam apenas restrições necessárias que descrevem a região viável. O conjunto formado por apenas estas restrições é denominado de representação mínima do espaço viável. Como a solução ótima do problema da PQ deve estar contida nesta representação mínima é interessante estabelecer condições para que se tenha esta. Para tanto, o teorema 2.III.4.1 de Obuchowska & Caron (1995) escrito para o caso da PQ pode ser usado.

Teorema 2.III.4.1: *(da região viável mínima da PQ)* - As restrições de um problema da programação quadrática descrevem uma região viável mínima se e somente se não existem restrições redundantes.

Assim, o algoritmo de conjuntos ativos deve ser tal a eliminar as restrições redundantes. Lucia & Xu (1990) propõe a seguinte estrutura para o algoritmo de conjuntos ativos.

1. Partindo-se de um conjunto ativo I_j , resolve-se (2.III.4.2) para a obtenção de d , λ , μ_{I_j} .
2. Se todos os componentes de μ_{I_j} forem positivos e se todas as restrições não ativas forem satisfeitas, d , λ , μ_{I_j} correspondem à solução do problema e então o algoritmo é finalizado.
3. Se existem i restrições ativas para as quais $\mu_i < 0$ então elimine aquela que tiver o maior valor em módulo de μ_i .

4. Verifique se existem restrições em A_I não pertencentes a I_j que são violadas. Incorpore em I_j todas as restrições violadas.
5. Se o número de restrições ativas resultante do procedimento dado em 4 for maior que a dimensão do problema proceda como segue:
 retire as primeiras i restrições de I_j que satisfazem $\max_{i \in I_j} \left| \tau = \frac{b_i}{A_i d} \right|$, de forma que se tenha $\dim(I_j) = n$, sendo n a dimensão de d .
6. Retorne ao passo 1.

Devemos salientar apenas que Lucia & Xu (1990) não indicam como restrições linearmente dependentes são tratadas. Ainda, quando o algoritmo é usado conjuntamente com o algoritmo SQP, uma fase de pré processamento é necessária para determinar o primeiro conjunto ativo. Mais recentemente, Lucia et ali (1996) efetuam alterações sobre o algoritmo ora apresentado. Discutem que a fase de pré processamento pode não ser necessária, procedimento, este, por nós verificado.

Na discussão apresentada, detivemo-nos à análise de inconsistências oriundas de linearizações nas restrições de desigualdade. Basicamente, vimos que estas eram devidas a se ter uma região viável a nível da PQ vazia e à presença de restrições redundantes. Com relação às restrições de igualdade, análises semelhantes podem ser feitas, i.e., podem existir restrições redundantes ou tais que o sistema de equações que se quer resolver seja impossível. No entanto, uma ligeira diferença existe em relação ao caso de se ter restrições inequações inconsistentes. Note, primeiramente, que o número de restrições de equações será no máximo igual à dimensão do problema. Logo para que o sistema de equações seja impossível devemos ter que A_E deve ser de posto de linha deficiente. Ainda, uma restrição só será redundante se ela for linearmente dependente das demais, ou seja, novamente restrições redundantes podem ser identificadas analisando-se o posto da matriz A_E . Estes resultados serão formalizados no capítulo 4. Assim, a uma primeira vista, tratar restrições de igualdade inconsistentes é mais simples, uma vez que a análise se resume em se detectar o posto da matriz e as restrições linearmente dependentes (LD) se houver alguma. Contudo, esta análise só é válida quando não existem restrições de desigualdade. No caso geral, em que se tenha A_E e A_I , um outro tipo de inconsistência pode surgir, a saber, restrições que se tornam redundantes ou quando a interseção dos espaços formados por A_E e A_I é vazia.

Assim, e.g., no algoritmo de conjuntos ativos descrito anteriormente, uma nova verificação que deve ser feita é com relação a uma eventual incompatibilidade entre estes dois tipos de restrições. Esta verificação deve ser traduzida por condições do tipo “se isto ocorrer proceda assim”, as quais podem se tornar custosas. Assim, percebe-se o interesse de se tratar estas inconsistências de uma forma mais sistemática.

Para que a discussão das inconsistências entre as restrições de igualdade e desigualdade seja facilitada, iremos introduzir o conceito de decomposição de variáveis. Basicamente, a idéia da decomposição advém do fato de que para problemas reais o número de graus de liberdade, é relativamente pequeno. Assim, usam-se as restrições de igualdade para se restringir o domínio de procura.

Considere que $[Z \ T]$ é uma base para o espaço formado pelas restrições de igualdade, tal que Z é uma base para o espaço nulo de A_E , ou seja, teremos que:

$$\begin{aligned} A_E Z &= 0 \\ A_E \bar{d} &= b_E \text{ e } \bar{d} = T\hat{d} \Rightarrow A_E d = b_E \text{ para } d = Zy + T\hat{d}; \forall y \in \mathbb{R}^{n-m} \\ Z &\in \mathbb{R}^{n \times n-m}; T \in \mathbb{R}^{n \times m} \end{aligned} \quad (2.III.4.4)$$

Em função disto percebemos que a variável d pode ser decomposta em termos de uma variável reduzida y e uma solução particular do sistema de equações, a saber, \bar{d} ou $T\hat{d}$. Assim, o problema da PQ que se deseja resolver passa a ser descrito como:

$$\min_{y \in X_r} \frac{1}{2} y^T Z^T H Z y + (\bar{d}^T H + c^T) Z y; X_r = \{y \mid A_I Z y \leq b_{Ir}; b_{Ir} = b_I - A_I \bar{d}\} \quad (2.III.4.5)$$

Assim, analisar inconsistências entre os diversos tipos de restrições pode-se traduzir na relação entre a base obtida Z e o conjunto formado pelas restrições de desigualdade A_I . Duas situações surgem, a de que a restrição inequação é redundante com pelo menos uma restrição em A_E e neste caso ela pode ser eliminada ou a de que ela é incompatível com a restrição de igualdade. Em qualquer uma destas situações particulares, o que ocorre, quando se efetua a decomposição, é que surgem linhas nulas na matriz $A_I Z$.

Para entendermos a extensão do problema que pode surgir, consideremos um caso particular em que $m=n$ e em que A_I é constituída apenas de restrições de limites nas variáveis. Para esta situação é fácil perceber que $A_I Z$ será nula em qualquer iteração do algoritmo SQP. Assim, o que ocorre é que as restrições de limites poderão ser ignoradas, embora possa existir uma solução viável. Neste caso, a base Z poderá imprimir uma direção de busca que não leva em conta o problema de otimização, i.e., visa-se tão somente a resolver o sistema de equações $h(x)=0$ pelo método de Newton. Apesar desta situação ser o caso extremo, ela serve para ilustrar o que ocorre quando $A_I Z$ é nula. Neste ponto, queremos comentar sobre o que não aparece citado em artigos da literatura. Schmid & Biegler (1994) utilizam uma formulação do problema (P1) ou (P2) tal que as restrições inequações surgem apenas como restrições de limites, ou seja, todas as restrições originais inequações são transformadas em equações. Os autores procedem desta forma, alegando que para muitos dos problemas da engenharia química, o número de graus de liberdade é pequeno e assim este procedimento se justifica, abordagem, esta, criticada por outros (Lucia et al, 1996). Sem entrar, por enquanto, no mérito deste procedimento, o que queremos ressaltar é que o problema de se ter linhas nulas em $A_I Z$ existe e a inclusão de uma variável artificial para se minimizar problemas de PQ inviáveis (a ser comentado mais adiante) não é capaz de contornar este problema. Apesar disto, os autores não citam como esta situação pode ser contornada sem o uso de uma fase de pré processamento.

Um outro caso particular de inconsistência é a situação em que o gradiente das restrições não lineares é um vetor nulo. Embora, este caso possa ser considerado como um caso particular das inconsistências descritas anteriormente, preferimos considerar como sendo um caso especial de inconsistência. Isto porque uma linha nula em A_E ou A_I faz com que na verdade esta restrição seja ignorada. Isto é, não há meios de se garantir que a direção de busca resultante do algoritmo de solução gere uma perturbação tal que o gradiente da restrição seja não nulo na próxima iteração. Muitos dos casos de falha de convergência de algoritmos SQP (entre outros métodos) está associado a este problema, alguns destes exemplos serão apresentados no capítulo 5 e apêndice 2.

Coleman (1984) apresenta diversas maneiras de se efetuar a decomposição, sugerindo basicamente quatro métodos principais de se obter as bases Z e T , a saber: o método das bases ortonormais, o método das bases ortogonais, o método de eliminação ou das bases

coordenadas e o método das transformações elementares. Destes, nos deteremos em pormenores apenas no primeiro, que é o método empregado pelo algoritmo MISQPSOL. Referências que tratam dos demais métodos podem ser vistas em Coleman (1984), Schmid & Biegler (1993, 1994).

Considere inicialmente que A_E seja de posto de linha pleno. Seja a seguinte fatoração QR:

$$A_E^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (2.III.4.6)$$

onde, $Q_1 \in IR^{n \times m}$; $Q_2 \in IR^{n \times n-m}$

A partir de (2.III.4.6), percebe-se que as bases Z e T podem ser obtidas como:

$$\begin{aligned} Z &= Q_2 \\ T &= Q_1 \end{aligned} \quad (2.III.4.7)$$

Ainda, a solução particular \bar{d} pode ser obtida, por exemplo, como:

$$\begin{aligned} R^T \tilde{y} &= b_E \\ \bar{d} &= T \tilde{y} \end{aligned} \quad (2.III.4.8)$$

Quando se tem A_E de posto de linha deficiente, usar (2.III.4.6) diretamente para a obtenção de Z e T por (2.III.4.7) não é adequado pois a matriz R será de posto deficiente e conseqüentemente (2.III.4.8) não pode ser resolvido. Ainda, se obtivermos Z de (2.III.4.7), teremos que a sua dimensão será maior que a real. Se o posto de A_E for mr então a dimensão de Z é dada por $n \times (n - mr)$. Na verdade (2.III.4.7), ainda, poderia ser usada para determinar Z , desde que apenas as últimas $n - mr$ colunas fossem consideradas. Contudo, o problema de se ter R singular persistiria e além disso seria necessário estabelecer mr .

Na verdade, em função da natureza da fatoração QR, estes problemas são facilmente contornáveis. Para tanto apresentaremos um teorema importante extraído de Stewart & Sun (1990).

Teorema 2.III.4.2: (da fatoração QR) - Seja $A \in \mathbb{R}^{o \times p}$, $o \geq p$ uma matriz de posto igual a pr . Então existe uma matriz de permutação P e uma matriz unitária Q , tais que:

$$Q^H A P = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

onde, R_{11} é triangular superior, não singular e de ordem pr . (O índice H indica ser a matriz transposta conjugada).

A partir deste teorema o seguinte corolário pode ser facilmente obtido.

Corolário 2.III.4.1: (da fatoração QR) - A fatoração QR da matriz transposta de $A_E \in \mathbb{R}^{m \times n}$, $n \geq m$ com posto $mr < m$ pode ser usada para a detecção de linhas LD na matriz A_E .

prova:

Pelo teorema 2.III.4.2 se considerarmos $P=I$ temos que $Q^T A_E = \begin{bmatrix} R \\ 0 \end{bmatrix}$,

sendo que R será triangular superior de ordem m , contudo poderá ser de posto deficiente. Assim, o que devemos estabelecer para provar o corolário é como a matriz P pode ser composta a partir de $P=I$ por meio de informações existentes em R de forma que a estrutura do teorema 2.III.4.2 seja obtida. A matriz R será de posto deficiente se e somente se existirem elementos nulos na diagonal principal. Assim, a formação de P deve ser feita como: obtenha o índice do primeiro elemento nulo da diagonal de R , digamos i . Permute a i ésima linha de

$$A_E \text{ e atualize a fatoração QR, ou seja, tem-se: } Q^T A_E P_1 = \begin{bmatrix} R_1 & R_2 \\ 0 & \end{bmatrix},$$

onde R_1 terá ordem $m-1$ e poderá ser singular. Agora, basta aplicar o mesmo procedimento para R_1 e ao final deste procedimento teremos chegado à estrutura apresentada no teorema 2.III.4.2.

O corolário 2.III.4.1 pode ser usado para a obtenção das bases Z e T , além da solução particular \bar{d} . Note que dependendo da dimensão da matriz A_E a obtenção de P pode ser custosa.

Schmid & Biegler (1994) comentam, ainda, que para muitos problemas o termo $Z^T Q \bar{d}$ pode ser desprezado pois \hat{d} deve convergir mais rapidamente a zero que y .

Neste ponto é interessante salientar a importância dos métodos de decomposição. O método SQP é considerado o mais eficiente para problemas de pequeno e médio porte. Para problemas grandes, modificações que visem a um aumento da eficiência computacional fazem-se necessárias, sendo esta uma área de pesquisa em aberto. Schmid & Biegler (1993) apontam duas escolas de pensamento distintas que surgiram para avaliar formas de tornar os algoritmos SQP eficientes para problemas grandes. Uma delas, preocupa-se basicamente em se buscar algoritmos eficientes de solução de problemas da PQ esparsos. Estudam-se métodos de atualização da matriz Hessiana eficientes, bem como esquemas de fatoração e atualização destes. A outra abordagem se baseia nos métodos de decomposição. Ou seja, efetuar ou não uma decomposição, define, na verdade, uma classe de algoritmos a que um dado código SQP pertence.

Quando se efetua a decomposição, surge a questão de como os multiplicadores de Lagrange das restrições de igualdade podem ser recuperados, uma vez que eles não surgem

na formulação (2.III.4.5). Pré multiplicando (2.III.4.2) por $\begin{bmatrix} Z & T & 0 \\ 0 & & I \end{bmatrix}^T$, obtém-se:

$$\begin{bmatrix} T^T H & T^T A_E & T^T A_I \\ Z^T H & 0 & Z^T A_I \\ A_E & 0 & 0 \\ A_I & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} -T^T c \\ -Z^T c \\ b_E \\ b_I \end{bmatrix} \quad (2.III.4.9)$$

Donde podemos perceber que λ pode ser obtido por intermédio de (2.III.4.10).

$$T^T H d + T^T A_E^T \lambda + T^T A_I^T \mu = -T^T c \quad (2.III.4.10)$$

Schmid & Biegler (1993) mostram que como valores exatos de λ só são necessários na solução ótima do problema e neste caso tem-se $d=0$, então (2.III.4.11) é uma boa aproximação para (2.III.4.10).

$$T^T A_E \lambda = -T^T A_I \mu - T^T c \quad (2.III.4.11)$$

Ainda, um comentário pertinente é com relação à relevância da decomposição quando do uso de algoritmos de pontos interiores. Aparentemente uma condição necessária para se aplicar alguns dos desenvolvimentos recentes é não se ter a nível da PQ restrições de igualdade (Obuchowska & Caron, 1995). Assim, alguns métodos de pontos interiores poderiam ser aplicados ao problema reduzido. Este é, por exemplo, o caso do algoritmo de Kojima et al (1991) que pode ser usado para a resolução do problema da PQ reduzido, desde que este seja consistente e que o Hessiano seja PD. Kyriakopoulou & Kalitventzeff (1997) apresentam um algoritmo SQP em que o problema da PQ é resolvido por um método de pontos interiores.

Embora a decomposição pareça ser bastante vantajosa, não são muitos os algoritmos da literatura que fazem dela uso. De fato, existem algoritmos que transformam restrições de igualdade em desigualdade como o de Heinz & Spellucci (1994). Este procedimento não nos parece ser adequado, especialmente se os autores aplicam um método de conjuntos ativos para a resolução do problema da PQ, ou seja, um algoritmo exponencial.

Um ponto importante que se deve comentar agora é com relação aos problemas que apresentam X_I vazios. Por exemplo, vimos que para o procedimento descrito em Lucia & Xu (1990) era necessário que o espaço viável da PQ fosse não vazio. Além disso, um espaço viável vazio significa que não existe solução e neste caso não existiria uma direção de busca o que não pode ocorrer. Vimos que esta situação surgia por termos uma linearização inconsistente, ou seja, por não linearidades no modelo. É de se esperar que estas sejam mais acentuadas quando se está longe da solução ótima de (P2) ou (P1), uma vez que é nestas situações que se espera ter d suficientemente grande a ponto de causar problemas nas novas linearizações. Assim, em função desta discussão duas idéias podem surgir inicialmente, a saber, a de se limitar d e a de se tolerar uma folga na descrição da região viável, ou seja, limites em b_E e b_I de modo a ampliar a região viável. Um exemplo em que a primeira idéia é abordada pode ser visto e.g. em Lucia & Xu (1990). Wright (1989) efetua ainda alguns outros comentários. Com relação à ampliação da região viável, diversas abordagens podem ser encontradas na literatura, como e.g. Bartholomew-Biggs &

Hernandez (1995), Schmid & Biegler (1994), Heinz & Spelluci (1994), Nemhauser (1989), Stoer (1985) ou Bazarra et al (1993). Como exemplos citamos as duas primeiras.

Bartholomew-Biggs & Hernandez (1995) propõem a seguinte estrutura:

$$\min_{d \in X_{lb}} \frac{1}{2} d^T H d + c^T d; X_{lb} = \left\{ d \mid A_E d = b_E - \rho \frac{\lambda - \lambda_k}{2}; A_I d \leq b_I - \rho \frac{\mu - \mu_k}{2} \right\} \quad (2.III.4.12)$$

onde, ρ é um parâmetro de penalidade, atualizado conforme a iteração do SQP, λ_k e μ_k se referem aos multiplicadores de Lagrange da iteração k do algoritmo, ou seja, correspondem à solução ótima da PQ calculada na iteração anterior.

Schmid & Biegler (1994) sugerem a seguinte estrutura:

$$\min_{y \in X_k} \frac{1}{2} y^T Z^T H Z y + (c^T + \bar{d}^T H) Z y + \rho \left(\xi + \frac{\xi^2}{2} \right) - \bar{d}^T H Z \xi; X_k = \{ y \mid l_r \leq Z y - \xi \bar{d} \leq u_r; \xi \geq 0 \} \quad (2.III.4.13)$$

onde, ρ é um parâmetro de penalidade, sendo igual a um número suficientemente grande, ξ é uma variável que mede a inviabilidade, l_r e u_r são os limites inferior e superior nas variáveis de decisão do problema da PQ reduzido.

Wright (1989) mostra que efetuar um relaxamento no conjunto X_I pode não afetar a convergência do algoritmo SQP. Neste contexto, ampliar a região viável, i.e., $Ax \leq b + \xi$, $\xi \geq 0$ é razoável. Ainda, Wright (1989) mostra que, na verdade, algumas restrições nem precisam ser satisfeitas numa dada iteração k , desde que elas não estejam ativas na solução ótima. Ou seja, diversas formulações do problema da PQ podem existir e a obtenção delas é mais dependente da análise de convergência do algoritmo e que particularidades um dado autor quer imprimir ao seu algoritmo.

A formulação do problema da PQ não é única. As características que este deve ter, são a inclusão de uma aproximação do Hessiano do Lagrangeano da função não linear (usa-se uma aproximação de 2ª ordem) e uma medida que quantifique a violação das restrições. Neste caso, pode-se proceder a uma aproximação linear de 1ª ordem e então (2.III.4.1) é obtido, e.g., com as definições em (2.III.4.3), ou ainda, as restrições podem ser embutidas na função objetivo como um fator de penalidade. Também, em função de como se deseja

tratar problemas da PQ inviáveis, alterações nas formulações básicas podem ser feitas, introduzindo-se um fator de relaxamento. Este pode ser apenas um termo de penalidade ou estar associado às estimativas dos multiplicadores de Lagrange. Ainda, vimos que efetuar uma decomposição do espaço das soluções pode ser vantajoso. Assim, percebemos que um outro ponto de distinção entre os algoritmos SQP, além da maneira de se obter a estimativa do Hessiano, é a própria formulação do problema da PQ. Ainda, a resolução completa de (2.III.4.1) pode não ser necessária (Wright, 1989). Novamente, a taxa de convergência local e a propriedade de convergência global irão depender das hipóteses assumidas nesta fase.

A resolução do problema da PQ pelo método de conjuntos ativos requer que se resolva sucessivamente sistemas de equações lineares. Se a matriz envolvida não for bem condicionada pode-se ter problemas numéricos, os quais são aumentados dependendo da precisão da máquina em que os cálculos são realizados. Para minimizar erros numéricos, escalonamentos nas variáveis e restrições são sugeridos.

Escalonamentos das restrições e função objetivo do problema (P1) ou (P2) podem ser vistas em e.g. Biegler & Cuthrell (1985), Heinz & Spellucci (1994) e Rijcaert & Walraven (1985), a saber:

$$\begin{aligned}\hat{f}(x) &= \text{diag}(\sigma_F) f(x) \\ \hat{h}(x) &= \text{diag}(\sigma_H) h(x) \\ \hat{g}(x) &= \text{diag}(\sigma_G) g(x)\end{aligned}\tag{2.III.4.14}$$

onde, $\sigma_F, \sigma_H, \sigma_G$ são vetores de dimensão compatível com as funções a que se referem.

Diversos valores podem ser escolhidos para os elementos da diagonal das matrizes de escalonamento. Não existe consenso para o valor que deve ser escolhido, sendo este mais dependente do problema e da experiência do usuário. Biegler & Cuthrell (1985) relatam referências que preferem escalonar as restrições, de modo que elas fiquem em torno de 1 e outras em que elas deveriam ser fixadas entre 10 e 100. Nossas experiências indicam que valores entre 0.1 e 100 podem ser bem tolerados mesmo trabalhando-se em aritmética de precisão simples. Ou seja, parece não haver consenso sobre valores ótimos para o escalonamento.

Ainda, pode ser bastante desejável efetuar um escalonamento das variáveis. Este pode até trazer benefícios em relação ao condicionamento numérico da matriz Hessiana. Um procedimento possível é seguir as sugestões de Rijcaert & Walraven (1985) e Gill et al (1985), a saber:

$$\begin{aligned}\hat{x} &= \text{diag}(\sigma_x)x \\ \sigma_x &\in \mathbb{R}^n\end{aligned}\tag{2.III.4.15}$$

Uma alternativa pode ser encontrada, ainda, em Biegler & Cuthrell (1985) e Papalambros & Wilde (1988), a saber:

$$\begin{aligned}\hat{x} &= \text{diag}(\sigma_x)x + \sigma_o \\ \sigma_x, \sigma_o &\in \mathbb{R}^n\end{aligned}\tag{2.III. 4.16}$$

Uma interpretação para (2.III.4.16) é adimensionalizar as variáveis segundo os seus limites finitos físicos, conforme a equação (2.III.4.17).

$$\hat{x}_i = \frac{x_i - l_i}{u_i - l_i} \rightarrow \sigma_{x,i} = \frac{1}{u_i - l_i}; \sigma_{o,i} = -\frac{l_i}{u_i - l_i}\tag{2.III.4.17}$$

sendo, l e u , respectivamente o limite superior e inferior em x .

Biegler & Cuthrell (1985) e Heinz & Spellucci (1994), ainda, efetuam comentários sobre que valores adotar para os vetores de escalonamento, os últimos, efetuando um novo escalonamento a cada iteração do algoritmo SQP.

Papalambros & Wilde (1988) efetuam diversas considerações com relação ao uso do método de conjuntos ativos para a resolução do problema da PQ. Os autores recomendam que apenas uma restrição seja eliminada do conjunto ativo por vez para que propriedades de convergência global do procedimento sejam garantidas. Na verdade, é por isso, que todas as implementações de métodos de conjuntos ativos para problemas da PQ eliminam apenas uma restrição por vez. A questão que surge é qual restrição deve ser primeiramente eliminada. A escolha é essencialmente empírica e muitas vezes escolhe-se a restrição que apresenta o maior valor em módulo do multiplicador de Lagrange. Heinz & Spellucci (1994)

eliminam a primeira restrição que contiver $\mu < 0$. Lucia & Xu (1990) eliminam a restrição com o maior valor em módulo.

A introdução de restrições violadas ao conjunto ativo também é um procedimento heurístico. Papalambros & Wilde (1988) sugerem a introdução da restrição mais violada. Lucia & Xu (1990) introduzem todas as restrições mais violadas possíveis enquanto que Heinz & Spellucci (1994) introduzem a restrição violada que tiver o menor índice. Heinz & Spellucci (1994), ainda, retiram todas as restrições que tiverem multiplicadores de Lagrange negativos antes de introduzir restrições violadas, procedimento, este, que difere ligeiramente de outras implementações.

Como os problema da PQ serão resolvidos iterativamente, empregar o procedimento de conjuntos ativos pode representar um custo demasiadamente elevado para se obter a solução da PQ. Quando se estiver próximo da solução ótima de (P1) ou (P2) é de se esperar que o conjunto ativo ótimo do problema da PQ não irá diferir muito de uma iteração para outra. Assim, é vantajoso fornecer ao problema da PQ quais as restrições que deverão estar ativas. Muitos dos algoritmos SQP fazem uso desta estratégia.

Como o problema (P1) ou (P2) representa uma situação real, então analisando-se este, pode-se ter uma idéia de quais restrições deverão estar ativas em sua solução ótima. Este poderia ser o caso da otimização em tempo real, em que o processo está operando e para o qual sabe-se que algumas variáveis devem seguir uma determinada direção. Para estes casos Papalambros & Wilde (1988) sugerem o emprego de técnicas de inteligência artificial para auxiliar a identificação do conjunto ativo ótimo.

Percebemos que o procedimento do método de conjuntos ativos, além de ser essencialmente empírico, é do tipo “se então, senão”, conjunto de regras estas que caracterizam os algoritmos exponenciais. Assim, o seu custo computacional pode ser elevado. O emprego de métodos duais-primais para a resolução da PQ pode então trazer alguns benefícios, embora estes também sejam algoritmos não polinomiais.

Uma outra questão que não foi abordada é com relação à solução do sistema de equações lineares. Basicamente, a cada iteração do algoritmo de conjuntos ativos, iremos resolver um sistema do tipo (2.III.4.18).

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix} \quad (2.III.4.18)$$

Então é de interesse resolver-se este sistema de uma forma adequada. Em seguida, iremos comentar algumas abordagens apresentadas na literatura.

A matriz à esquerda em (2.III.4.18) pode ser fatorada segundo Coleman (1984) como:

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} H & 0 \\ A & -AH^{-1}A^T \end{bmatrix} \begin{bmatrix} I & H^{-1}A^T \\ 0 & I \end{bmatrix} \quad (2.III.4.19)$$

Definindo-se a matriz M conforme a equação (2.III.4.20), a resolução de (2.III.4.18) pode ser feita resolvendo-se (2.III.4.21) e (2.III.4.22).

$$M = -AH^{-1}A^T \quad (2.III.4.20)$$

$$\begin{cases} Ht = -c \\ At + M\lambda = b \end{cases} \quad (2.III.4.21)$$

$$t = d + H^{-1}A^T\lambda \quad (2.III.4.22)$$

Um primeiro comentário surge neste ponto. Se H é singular, o procedimento não pode ser aplicado, tal como foi descrito. No capítulo 3, iremos discutir como estas equações podem ser usadas para esta situação. A resolução de (2.III.4.20) a (2.III.4.22) pode ser feita por intermédio de fatorações convenientes. Por exemplo se H é PD, então a fatoração de Cholesky pode ser usada. Em várias referências cita-se o uso das fatorações QR e de Cholesky para a resolução de (2.III.4.18).

Papalambros & Wilde (1988), ou ainda, Schmid & Biegler (1994) interpretam a resolução de (2.III.4.18) num contexto mais geral que se baseia nas recomendações de Fletcher (1987). Note que (2.III.4.21) e (2.III.4.22) pode ser expresso em notação matricial como (2.III.4.23):

$$\begin{bmatrix} d \\ \lambda \end{bmatrix} = \begin{bmatrix} -G & D \\ D^T & -U \end{bmatrix} \begin{bmatrix} -c \\ b \end{bmatrix} \quad (2.III.4.23)$$

sendo que G , D e U se relacionam com o problema original para o caso em que H seja não singular e A de posto de linha pleno como:

$$\begin{aligned} G &= H^{-1} - H^{-1}A(A^TH^{-1}A)^{-1}HA^TH^{-1} \\ D &= H^{-1}A(A^TH^{-1}A)^{-1} \\ U &= -(A^TH^{-1}A)^{-1} \end{aligned} \quad (2.III.4.24)$$

Diferenças entre as implementações de diferentes códigos podem ser entendidas como maneiras distintas de se resolver (2.III.4.24). A vantagem da representação (2.III.4.24) é que esquemas mais simples de atualização de matrizes podem ser obtidos, i.e., faz-se uso de fórmulas de recorrência para se atualizar as matrizes em questão.

Se H for PD então uma abordagem possível é a de se obter seus fatores de Cholesky, L , e conjuntamente com uma fatoração QR de $L^{-1}A$ teríamos como pode ser visto, e.g., em Schmid & Biegler (1994):

$$H^{-1}A = L^{-1^T}Q \begin{bmatrix} R \\ 0 \end{bmatrix}; \quad QR = L^{-1}A \quad (2.III.4.25)$$

As equações (2.III.4.21) e (2.III.4.22) serão o ponto de partida para o esquema que definirá a maneira de se atualizar as matrizes constantes em (2.III.4.24) e conseqüentemente como o sistema (2.III.4.23) será resolvido. Gill et ali em Nemhauser (1989) efetuam recomendações semelhantes, particularmente com relação ao uso da fatoração descrita em (2.III.4.25) e citam inúmeras referências que descrevem como esta expressão pode ser adequadamente usada.

Schmid & Biegler (1994) ainda comentam que a partir de (2.III.4.25) pode-se chegar em:

$$\begin{aligned}
 L^{-1^T} Q &= L^{-1^T} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} = B \\
 G &= B_2 B_2^T \\
 D &= B_1 R^{-1^T} \\
 U &= -R^{-1} R^{-1^T}
 \end{aligned}
 \tag{2.III.4.26}$$

A partir de (2.III.4.26) fórmulas recursivas para o cálculo de d_k e os multiplicadores de Lagrange podem ser construídas, algumas das quais se relacionam com a formulação do problema dual.

Schmid & Biegler (1994) efetuam alterações sobre o algoritmo de Goldfarb & Idani (1983) propondo uma forma mais eficiente de se atualizar o conjunto ativo. Não julgamos necessário efetuar mais comentários, pois as propostas deles não tem relação direta com o nosso algoritmo.

No procedimento do método de conjuntos ativos, o conjunto ativo poderá diferir pouco de uma iteração a outra. Assim, efetuar-se novas fatorações a cada iteração do algoritmo é desnecessário e técnicas de atualização de fatorações devem ser usadas. Isto é ainda mais crítico quando se têm problemas grandes e esparsos. Coleman (1984) sugere alguns procedimentos de como as atualizações podem ser feitas.

Uma outra questão que pode surgir é com relação à necessidade de se resolver exatamente o problema da PQ. Como este representa uma aproximação do problema não linear, então, nas primeiras iterações do algoritmo SQP, é de se esperar que a linearização seja bastante diferente daquela obtida na solução ótima. Assim, pode ser vantajoso não se resolver completamente o problema da PQ ou então não se atualizar a cada iteração a matriz Hessiana. Varvarezos et ali (1994) comentam que a resolução incompleta do problema da PQ pode não afetar a convergência do algoritmo SQP.

Varvarezos et ali (1994) formulam o problema da PQ como sendo um problema em que existem apenas restrições de igualdade e de limites. Aplica-se, ainda, um procedimento de decomposição. Assim, as restrições do conjunto ativo serão apenas as de limites. As sugestões dos autores se referem à resolução de um problema da PQ incompleto. Considere que se tenha obtido uma dada direção de busca d_k , a esta direção de busca verifica-se qual a

maior violação nas restrições obtendo-se o parâmetro $\vartheta = (d_k - d_v)_i$, onde d_v corresponde ao limite da máxima violação. A partir deste parâmetro, segue-se um procedimento heurístico para se verificar se uma nova iteração no método de conjuntos ativos será realizada.

A figura 2.III.4.2 mostra o procedimento sugerido por Varvarezos et ali (1994).

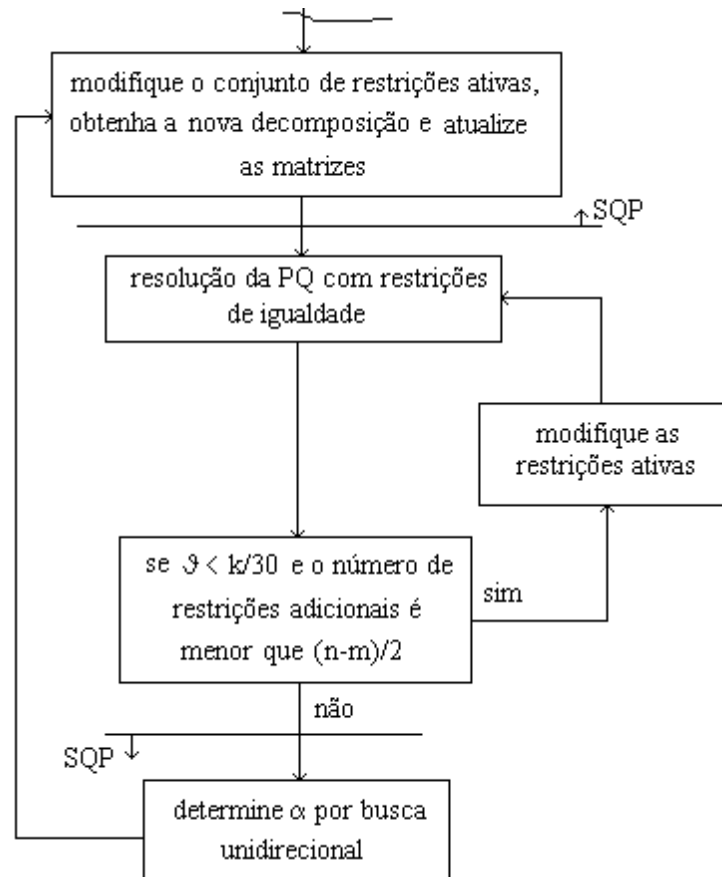


Figura 2.III.4.2: Estratégia de Varvarezos et ali (1994) para aumentar a eficiência do algoritmo SQP

Wright (1989) mostra que determinados relaxamentos na resolução do problema da PQ não alteram a propriedade de convergência local do método SQP. Por exemplo, se X_l não contiver restrições que se sabem não estarem ativas na solução ótima do problema, então a sequência gerada pelo algoritmo convergirá para a solução ótima de (P1) ou (P2). Restrições de igualdade e até mesmo inequações podem não ser satisfeitas com exatidão e mesmo assim a convergência ainda é assegurada. Por exemplo, podemos, ainda, ter X_l formado como: $X_l = \left\{ d \mid A_E d = b_E; A_{I_j} d = b_{I_j} \right\}$, onde I_j corresponde a um conjunto de

restrições que se espera estarem ativas e I_j pode variar finitamente. Não se resolver o problema da PQ exatamente pode ser interessante para problemas de dimensão maior, em que o custo em se obter d_k por (2.III.4.1) pode se tornar proibitivo.

Na discussão que segue iremos considerar que o problema da PQ apresenta apenas restrições de desigualdade. Esta não é uma hipótese restritiva, uma vez que basta aplicar-se um algoritmo de decomposição e a forma reduzida é obtida. Note, apenas que incompatibilidades entre as restrições de igualdade e desigualdade devem ser tratadas.

$$\begin{aligned} w &= q + Mz \\ w, z &\geq 0 \\ w^T z &= 0 \end{aligned} \tag{2.III.4.27}$$

A partir da transformação do problema da PQ em um problema linear complementar (PCL), cuja forma padrão é do tipo (2.III.4.27), poder-se-ia querer aplicar, e.g., um algoritmo de pontos interiores. Neste momento, alguns inconvenientes podem surgir. Kojima et al (1991) efetuam uma análise pormenorizada de uma classe de algoritmos de pontos interiores para a solução do PCL. No entanto, estes algoritmos só podem ser utilizados quando se impõe algumas restrições sobre a matriz M em (2.III.4.27). A condição menos restritiva discutida pelos autores é a de que M deve pertencer à classe CS de matrizes. Esta condição pode ser restritiva para a classe dos problemas da PQ considerados na solução de (P1) ou (P2) pelos algoritmos SQP. Isto advém do fato de que podemos ter problemas da PQ inconsistentes e neste caso pode não se conseguir obter uma solução do problema ou, ainda, o algoritmo pode não ser aplicado. Outras aplicações de algoritmos de pontos interiores para problemas da PQ podem ser encontrados. Han et al (1990) discutem o uso de um tipo de algoritmo de pontos interiores para um problema da PQ com restrições de limites apenas. Em princípio, esta poderia não ser uma hipótese restritiva, já que qualquer problema da PQ pode ser transformado neste tipo de problema efetuando-se uma decomposição de variáveis e transformando-se restrições de desigualdade em igualdade. Dois problemas no entanto surgem. A matriz H deve ser PSD, o que pode levar a uma restrição desnecessária. Além disso, se A_E não for de posto pleno, pode-se não obter a procurada representação, além dos problemas de inviabilidade. Estes problemas existem também para os algoritmos de conjuntos ativos, só que ainda, parecem

não ter sido alvo de grandes estudos na utilização em algoritmo de pontos interiores conjuntamente com o método SQP. Assim, apesar de parecerem ser bastante promissores, muito ainda deve ser pesquisado, para o uso destes algoritmos. Note que o caso geral do problema da PQ pode ser não convexo e conseqüentemente podemos ter problemas da PQ NP completos. Nestas situações, nenhum algoritmo de pontos interiores ou outro polinomial pode ser aplicado.

Além destes métodos, outros poderiam ser citados para a solução da PQ. Por exemplo, o uso do método de Keller, exposto em Dantzig (1963), o método do gradiente reduzido, ou o método da projeção do gradiente, procedimentos iterativos para a resolução do PCL, como os métodos de Jacobi, Gauss-Seidel ou SOR. Nenhuma destas alternativas, no entanto, parece ser atrativa. Bazaraa et ali (1993) discutem outras possibilidades, incluindo a solução da PQ por PCL para problemas de programação não convexa.

III.5. Definição da função de mérito e o problema de busca unidirecional

Para a garantia da convergência global do algoritmo, pode ser interessante que se restrinja o incremento realizado nas variáveis de uma iteração a outra do algoritmo SQP. A diminuição no incremento realizado deve ser feita como uma medida de se evitar um aumento de inviabilidade nas restrições não lineares. Desta forma, a obtenção desta poderia ser feita minimizando-se uma função penalidade, que, por outro lado, teria como mínimos locais os mesmos do problema que se quer resolver. Ainda, quando se está próximo da solução ótima, não se atenuar a direção de busca é recomendável, para que se tenha taxas de convergência adequadas. Assim, a obtenção deste fator de atenuação na direção de busca, α_k pode ser feito a partir do seguinte procedimento de busca unidimensional:

$$\min_{\alpha_k} \Psi(x_k, d_k, \alpha_k, \lambda_k, \mu_k, \lambda_{k+1}, \mu_{k+1}, \eta_k) \quad (2.III.5.1)$$

sendo, $\Psi : IR \rightarrow IR$ a função de mérito e η_k um fator de penalidade.

Stoer (1985) indica que as características desejáveis da função de mérito deveriam compreender as seguintes:

- a direção de busca d_k obtida na solução de (2.III.4.1) deve ser uma direção descendente de Ψ
- é desejável que a função Ψ seja diferenciável, uma vez que minimizar (2.III.5.1) quando Ψ é não diferenciável em alguns pontos não é uma tarefa fácil
- x^* deve ser uma solução ótima de (2.III.5.1) para valores adequados de η_k
- Se x_k está suficientemente próximo de x^* então seria desejável que $\alpha_k=1$ para valores adequados de η_k

Deve ser ressaltado que o uso da função de mérito faz com que o algoritmo SQP convirja para uma solução ótima de Ψ , quando a convergência ocorre. No entanto, qualquer função de mérito quando usada em problemas suficientemente complexos pode ter soluções ótimas, independentemente do valor de η_k , que não são soluções de (P2), e muitas vezes, nem sequer são pontos viáveis de (P2) (Stoer, 1985). Um indicativo disto pode ser percebido se α_k decresce mais e mais a cada iteração, sem que ocorra um decréscimo da função objetivo ou quando x_k não se encontra dentro da região viável (Stoer, 1985).

Stoer (1985), ainda, efetua um apanhado dos desenvolvimentos feitos até então, apresentando algumas funções de mérito. A escolha da melhor função de mérito parece ainda ser uma questão sem solução. Isto talvez seja evidenciado pela grande variedade de funções usadas na literatura. É importante destacar que o papel da função de mérito pode ser fundamental para que se tenha convergência global do algoritmo. Para efeitos de exemplificação consideraremos as funções consideradas por Biegler & Cuthrell (1985) e por Bartholomew-Biggs & Hernandez (1995) apresentadas nas equações (2.III.5.2) e (2.III.5.3), respectivamente. Observe-se, apenas, que estas funções não são diferenciáveis em todo o domínio das variáveis de decisão. Outros exemplos podem ser encontrados, e.g., em Powell & Yuan (1986) ou Stoer (1985).

$$\Psi(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)_+ + \frac{\eta_k}{2} \|g(x)_+, h(x)\|^2 \quad (2.III.5.2)$$

$$g(x)_+ = \max[0, g(x)]$$

sendo, que η_k é calculado a cada iteração e x , λ , μ podem ser avaliados como:
 $x = x_k$ ou $x = x_k + \alpha_k d_k$, $\lambda = \lambda_k$ ou $\lambda = \lambda_k + \alpha_k (\lambda_{k+1} - \lambda_k)$, $\mu = \mu_k + \alpha_k (\mu_{k+1} - \mu_k)$ ou
 $\mu = \mu_k$.

$$\Psi(x, \lambda, \mu) = f(x) + \frac{1}{\eta_k} \left(\sum_{i=1}^m \left(h_i(x) - \frac{\eta_k \lambda_i}{2} \right)^2 + \sum_{i=1}^p \left(\max \left[0, g_i - \frac{\eta_k \mu_i}{2} \right] \right)^2 \right) \quad (2.III.5.3)$$

sendo, que η_k é calculado a cada iteração e x , λ , μ são definidos como no caso anterior.

Uma vantagem do método SQP é que valores exatos de λ_{k+1} , μ_{k+1} correspondentes a x_{k+1} obtido a partir de d_k não são necessários (Papalambros & Wilde, 1988). Assim, a busca unidirecional não precisa ser calculada com exatidão. Uma solução simplificada habitualmente usada, é aplicar a desigualdade de Armijo (1966) para a resolução do problema de busca unidirecional, i.e. aplica-se a equação (2.III.5.4) para a obtenção de α_k :

$$\Psi(x_k + \alpha_k d_k, \lambda_k + \alpha_k (\lambda_{k+1} - \lambda_k), \mu_k + \alpha_k (\mu_{k+1} - \mu_k)) - \Psi(x_k, \lambda_k, \mu_k) \leq \frac{1}{2} \alpha_k \nabla \Psi(x_k, \lambda_k, \mu_k) \begin{bmatrix} d_k^T & (\lambda_{k+1} - \lambda_k)^T & (\mu_{k+1} - \mu_k)^T \end{bmatrix}^T \quad (2.III.5.4)$$

A maneira de se calcular α_k a partir de (2.III.5.4) varia pouco de um algoritmo SQP para outro. Assim, iremos apresentar basicamente o procedimento de Biegler & Cuthrell (1985). Tamura & Kobayashi (1991) propõem um esquema de cálculo de α_k mais eficiente do ponto de vista computacional. Cabe, apenas comentar que quando a direção de busca obtida é descendente, α_k deve ser positivo. É por isso que a imensa maioria dos algoritmos SQP da literatura resolvem o problema de busca unidirecional incluindo a restrição de que α_k deve ser positivo. No entanto, mesmo quando H_k é mantida PD, uma direção não descendente em relação ao problema original da PNL pode ser gerada e nestes casos α_k deve ser negativa (Lucia et al, 1996). Outras complicações podem surgir se α_k for forçado a ser positivo quando da resolução do conjunto de restrições de igualdade pelos métodos de decomposição. Basicamente, a direção de Newton obtida pelo algoritmo de decomposição pode ser incompatível com as restrições inequações e para estas situações permitir que α_k seja negativo pode ser suficiente para que estas dificuldades sejam sanadas.

Procedimento de Biegler & Cuthrell (1985):

1. Inicie $\alpha_k=1$, $\beta=0.1$
2. $\eta = \max[0, \eta_{dd} + 10^{-3}]$
3. Se $\Theta(\alpha_k) > 0$ ou $\eta_{LS} > \eta_{dd}$ vá para 5
4. Se $\Psi(x_k + \alpha_k d_k) \leq \Psi(x_k) + \alpha_k \beta \nabla \Psi(x_k) d_k$ vá para 5, senão determine um novo α_k por ajuste quadrático de $\Psi(x_k, \alpha_k, d_k, \lambda_k, \mu_k, \lambda_{k+1}, \mu_{k+1})$ e vá para 3.
5. $x_{k+1} = x_k + \alpha_k d_k$

onde,

$$\left\{ \begin{array}{l} \eta_{dd} = \frac{\nabla f(x_k)^T d_k + (\mu_{k+1} - 2\mu_k)^T g(x_k)_+ + (\lambda_{k+1} - 2\lambda_k) h(x_k)}{\|g(x_k)_+, h(x_k)\|^2} \\ \eta_{LS} = \frac{\Psi(x_k + \alpha_k d_k, \lambda_k + \alpha_k (\lambda_{k+1} - \lambda_k), \mu_k + \alpha_k (\mu_{k+1} - \mu_k)) - \Psi(x_k, \lambda_k, \mu_k) - \alpha_k \beta \nabla \Psi(x_k)^T \begin{bmatrix} d_k \\ \lambda_{k+1} - \lambda_k \\ \mu_{k+1} - \mu_k \end{bmatrix}}{\Theta(\alpha_k)} \end{array} \right.$$

sendo, $\Theta(\alpha_k) = -\frac{1}{2} \|g(x_k + \alpha_k d_k), h(x_k + \alpha_k d_k)\|^2 + (\frac{1}{2} - \alpha_k \beta) \|g(x_k)_+, h(x_k)\|^2$ e λ_{k+1} e μ_{k+1} definidos como em (2.III.5.2).

III.6 Critérios de parada

Diversos critérios existem para se identificar se uma dada solução é ótima ou não. Um dos mais usados se baseia na idéia de que na solução ótima, a direção de busca obtida na programação quadrática será nula e, desta forma, se pequenos incrementos são feitos nas variáveis de decisão é porque se está perto da solução ótima. Assim, a idéia é medir o quanto as variáveis diferem de uma iteração a outra. Palomares & Mangasarian (1976), Papalambros & Wilde (1988) sugerem que a medida seja tomada como:

$$\|x_k - x_{k-1}\| \leq \epsilon_p; \epsilon_p > 0 \quad (2.III.6.1)$$

Outra sugestão pode ser vista em Edgar & Himmelblau (1989), a saber:

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\| - 1} < \varepsilon_p; \|x_k\| \neq 1; \varepsilon_p > 0 \quad (2.III.6.2)$$

No entanto a forma com que (2.III.6.2) é apresentada em Edgar & Himmelblau (1989) não é adequada, uma vez que existe uma indefinição para o caso de x ser unitária. Uma modificação possível de ser feita é apresentada em (2.III.6.3) para o caso de se ter x unitário.

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\| - 1 + \varepsilon_z} < \varepsilon_p; \varepsilon_p, \varepsilon_z > 0 \quad (2.III.6.3)$$

Outras medidas que quantifiquem o desvio da condição ótima podem ser tomadas com relação às condições de otimalidade. Da condição de Karush-Kuhn-Tucker, temos que o gradiente do Lagrangeano deve ser nulo na solução ótima, assim, ao invés de se considerar a equação (2.III.6.1) ou (2.III.6.2) ou (2.III.6.3), poder-se-ia fazer uso da equação (2.III.6.4), como feito por Lucia & Xu (1990) e também sugerido em Papalambros & Wilde (1988).

$$\|\nabla f(x_k) + \lambda_k^T \nabla h(x_k) + \mu_k^T \nabla g(x_k)\| \leq \varepsilon_p; \varepsilon_p > 0 \quad (2.III.6.4)$$

Uma outra questão importante é com relação à viabilidade do problema. Por exemplo (2.III.6.1) a (2.III.6.3) nada dizem com relação a estarem as restrições satisfeitas. Ou ainda, (2.III.6.4) nada afirma com relação ao valor dos multiplicadores de Lagrange. Assim, o uso isolado destas equações pode não ser satisfatório, uma vez que pode-se eventualmente encontrar uma solução inviável ou que de fato não seja ótima. Uma medida da inviabilidade pode ser feita de acordo com a equação (2.III.6.5) e uma verificação do valor dos multiplicadores de Lagrange pode ser feita pela equação (2.III.6.6).

$$\left\| \begin{bmatrix} h(x_k) \\ g_I(x_k) \end{bmatrix} \right\| \leq \varepsilon_v; \varepsilon_v > 0; I = \{i \mid g_i(x_k) = 0\}; g_J \leq 0; J = \{i \mid i \notin I\} \quad (2.III.6.5)$$

$$\mu_k \geq 0 \quad (2.III.6.6)$$

O uso de (2.III.6.5) é na verdade bastante recomendado. Um problema que pode surgir com relativa frequência, ao contrário do que se poderia pensar, é o fato de termos gradientes das restrições nulos. Nesta situação, o que ocorre é que a nível da PQ estas restrições poderão ser ignoradas e a direção de busca obtida pode não ocasionar uma perturbação tal que mude o valor do gradiente e neste caso as restrições não serão satisfeitas. Ainda, um outro problema que pode surgir, especialmente quando do uso de um procedimento de decomposição na resolução da PQ é termos restrições de desigualdade reduzidas nulas. Neste caso, elas também poderão ser ignoradas e não satisfeitas.