



UNIVERSIDADE ESTADUAL DO CEARÁ
FRANCISCO DALADIER MARQUES JÚNIOR

**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA
AVALIAR O DESEMPENHO DE ISPs, ATRAVÉS DA
ANÁLISE ENVOLTÓRIA DE DADOS (DEA)**

FORTALEZA – CEARÁ
2010

FRANCISCO DALADIER MARQUES JÚNIOR

**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA AVALIAR O
DESEMPENHO DE ISPs, ATRAVÉS DA ANÁLISE ENVOLTÓRIA DE
DADOS (DEA)**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual do Ceará - UECE como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Jorge Luiz de Castro e Silva

Co-orientador: Prof. Dr. Antônio Clécio Fontelles Thomaz.

FORTALEZA – CEARÁ

2010

M357e

Marques Júnior, Francisco Daladier

Estimação de uma fronteira eficiente para avaliar o desempenho de ISPs, através da Análise Envoltória de Dados/ Francisco Daladier Marques Júnior. Fortaleza, 2010.

107 p.

Orientador: Prof. Dr. Jorge Luiz de Castro e Silva

Dissertação (Mestrado Acadêmico em Ciência da Computação) – Universidade Estadual do Ceará. Centro de Ciência e Tecnologia.

1. Análise Envoltória de Dados (DEA). 2. Programação Linear. 3. Avaliação de Desempenho. 4. Redes de Computadores. 5. ISPs. I.Universidade Estadual do Ceará. Centro de Ciência e Tecnologia.

CDD:001.6

FRANCISCO DALADIER MARQUES JÚNIOR

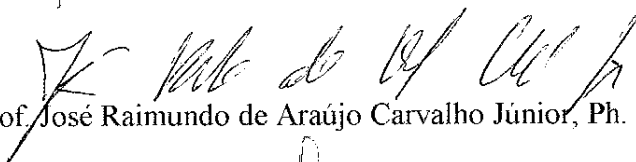
**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA AVALIAR O
DESEMPENHO DE ISPs, ATRAVÉS DA ANÁLISE ENVOLTÓRIA DE
DADOS (DEA)**

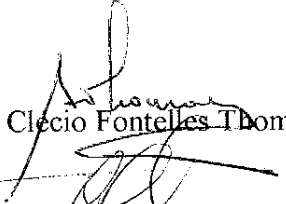
Dissertação de Mestrado submetida à comissão examinadora do Mestrado Acadêmico em Ciência da Computação, da Universidade Estadual do Ceará – UECE, como requisito parcial para obtenção de título de Mestre em Ciência da Computação. Aprovada pela comissão examinadora abaixo.

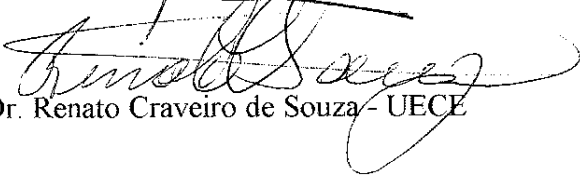
Aprovada em: 10/08/2010.

BANCA EXAMINADORA


Prof. Dr. Jorge Luiz de Castro e Silva (Orientador) – UECE


Prof. José Raimundo de Araújo Carvalho Júnior, Ph. D./ UFC


Prof. Dr. Antonio Clecio Fontelles Thomaz – UECE


Prof. Dr. Renato Craveiro de Souza - UECE

Agradecimentos

Em primeiro lugar agradeço a Deus pela força, discernimento e perseverança de lutar.

À minha esposa Maísa, pelas noites mal dormidas sem a minha presença, pelo compreensão, pelas palavras de incentivo, atenção, amor e carinho onipresentes;

Aos meus pais, Neide e Daladier, irmãos: D'Sávio e Raphaella e Mãe Corrinha que sempre me incentivarem a trilhar passos mais altos e me encorajaram a nunca desistir dos objetivos;

Aos meus avós, pelo carinho, apoio e orações. Maternos: vovó Santú (*in memorian*) e vovô Chico; Paternos: vovó Belinha e vovô Juarez (*in memorian*);

Aos primos e tios, pela preocupação com um futuro próspero em todos os pontos de vista;

Aos meus cães, Mayllow e Suzzy (*in memorian*), pelos momentos de alegria, amizade, dedicação, amor e companheirismo.

Ao meu orientador, Prof. Jorge Luiz, que depositou em mim a confiança antes esquecida. A ele agradeço a oportunidade de trilhar um caminho científico mais direcionado;

Ao meu co-orientador, Prof. Clécio Fontelles, pela dedicação e pelos ensinamentos sobre a DEA;

Ao IFPB, por minha liberação e investimento realizado;

Aos meus companheiros de apartamento que, na verdade, não foram apenas colegas, mas amigos de todas as horas a quem passo a citar: Alex e, em especial Walisson e Marcelo. Aos meus colegas Ênyo, pelo apoio e conselhos constantes; Marçal, pela alegria e descontração; Vigno; Fabiano, pelo comprometimento e seriedade; Gilzamir, além de outros;

Aos professores do MACC, em especial aos professores Marcial, Gustavo, Jerffeson, Valdísio, Celestino, André Ribeiro e André Moura, pelo comprometimento e lições em todos os sentidos;

Ao pessoal da secretaria e de apoio do MACC, em especial ao Marcos, Wágner, Elaine e o Auricélio (baixinho da limpeza);

Aos meus amigos e incentivadores no cotidiano: Fábio Gomes do IFPB, meu primo Osvaldo, meu amigo e veterinário Neto, meu colega de trabalho e amigo Ágio, Igor, Tiano, Samara, Fabiano (Fabinho) e Aninha;

Enfim, agradeço a todos que, de uma forma ou de outra torceram pelo meu sucesso nesta caminhada.

Resumo

A Análise Envoltória de Dados (DEA – *Data Envelopment Analysis*) é um método de estimação de fronteira Pareto-Koopmans eficiente, não-paramétrica, que auxilia tomadores de decisão a avaliarem Unidades de Tomada de Decisão (DMUs – *Data Making Units*) ou firmas sob uma ótica de estimação de pontos de referência, baseados na eficácia e produtividade. DEA tem sido largamente utilizada, nas últimas três décadas, na avaliação de escolas, setores de empresas, órgãos governamentais, empresas, organizações não-governamentais, entre outros para auxiliar a estas minimizarem a quantidade de insumos (entradas), a fim de produzir a mesma quantidade de produtos ou maximizar a quantidade de produtos (saídas), sem alterar a quantidade de insumos, através de um modelo de programação matemática ao solucionar um Problema de Programação Linear (PPL).

Ao analisar os ISPs (*Internet Services Providers* – Provedores de Serviço de Internet), nota-se que estes têm oferecido aos seus clientes, sub-serviços, principalmente pela falta de produtividade e eficácia nos processos empregados. Contudo, esta dissertação vislumbra contribuir, pontualmente, para que os ISPs possam oferecer melhores serviços aos seus clientes, bem como buscar reduzir vícios, com o intuito de obterem uma maior rentabilidade, ao ser aplicado a DEA, no julgamento de algumas variáveis, levantadas pontualmente, para solucionar tais problemas.

Palavras-Chave: Avaliação de desempenho. DEA. ISPs. Programação Linear. Redes de computadores.

Abstract

DEA (Data Envelopment Analysis) is a non-parametric estimation method of efficient Pareto-Koopman frontier which help the decision makers evaluate Data Making Units (DMUs) or firms under estimation from benchmarks based on efficiency and productivity. DEA have been broadly used in the past three decades in assessment from schools, enterprise sectors, governments, enterprises, NGO's (Non-Governmental Organization), and others to assist to minimize the inputs quantities to produce the same output quantities or to maximize the output quantities without changing the input quantities through of the mathematical programming to solve a Linear Programation Problem.

To analyze ISPs (Internet Service Providers) note that they offer their customers sub-services, due to lack of productivity and efficiency in the processes used. Hence, this dissertation sees occasionally contribute to the ISP's can offer better services to their customers, and seek to reduce defects in order to obtain a higher return when applied to the DEA in the trial of some variables raised promptly to resolve such problems.

Keywords: DEA. Computer Networks. ISPs. Linear Programming. Performance Evaluation.

Lista de Figuras

Figura 2.1. Espaço de busca de soluções do Simplex.	10
Figura 2.2. Envoltório de Dados	15
Figura 2.3. Medidas de eficiência e folgas de entrada [COELLI 1996].	15
Figura 2.4. Fronteiras CCR x BCC [COOPER et. al. 2007].	20
Figura 3.1. Mapa dos estados do Ceará e Paraíba com as cidades dos ISPs analisados.	26
Figura 3.2. Arquitetura proposta pelo trabalho.	29
Figura 3.3. Fluxograma algorítmico para estimação de uma fronteira eficiente DEA.	31
Figura 3.4. Fórmulas do modelo CCR/M/I aplicados a uma das DMUs analisadas.	32
Figura 3.5. Relatório da resolução de problemas de programação linear com o LINDO.	33
Figura 3.6. Fórmulas do modelo CCR/M/O aplicados a uma das DMUs analisadas.	34
Figura 3.7. Fórmulas do modelo CCR/E/I aplicados a uma das DMUs analisadas.	34
Figura 3.8. Fórmulas do modelo MBF aplicados a uma das DMUs analisadas.	37
Figura 3.9. Relatório das folgas e faltas do MBF relativo a uma das DMUs analisadas.	37
Figura 4.1. Saída do software proposto para o CCR/E/I.	41
Figura 4.2. Parceiros de Referência (<i>Benchmrks</i>) das DMUs.	41
Figura 4.3. Sumário de Eficiências Técnicas das DMUs calculadas no Win4DEAP.	42
Figura 4.4. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas.	42
Figura 4.5. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas.	42
Figura 4.6. Gráfico das Eficiências Técnicas do modelo CCR gerado pelo Solver DEA proposto.	43
Figura 4.7. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.	43
Figura 4.8. (a) Sumário das faltas das saídas e (b) Sumário das folgas das entradas, segundo o CCR/E/O.	44

Figura 4.9. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o CCR/E/O.	44
Figura 4.10. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/I.	45
Figura 4.11. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas, segundo BCC/E/I.	45
Figura 4.12.(a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/O.	46
Figura 4.13. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o BCC/E/O.	46
Figura 4.14. Gráfico das Eficiências Técnicas do modelo BCC gerado pelo <i>Solver</i> DEA proposto.	47
Figura 4.15. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.	48
Figura 4.16. Resultados do MBF calculados com o <i>Solver</i> DEA proposto.	48

Lista de Tabelas

Tabela 3.1. Entradas com os dados originais dos ISPs analisados.....	27
Tabela 3.2. Saídas com os dados originais dos ISPs analisados	27
Tabela 3.3. Matriz de correlação dos dados originais.	27
Tabela 3.4. Matriz de correlação definitiva dos dados das DMUs.....	31
Tabela 3.5. Dados das DMUs normalizados.....	32
Tabela 4.1. Ranqueamento das DMUs analisadas segundo CCR/E/I usando o DEA-Solver da Saitech.	43
Tabela 4.2. Eficiências técnicas, de escala e RVE's das DMUs analisadas.	47
Tabela 4.3. Ranqueamento das DMUs analisadas segundo BCC/E/I usando o DEA-Solver da Saitech.	47

Lista de Siglas e Abreviaturas

BCC	Barnes, Charnes e Cooper
BCC/E/I	BCC, modelo de envoltório e orientação à entrada
BCC/E/O	BCC, modelo de envoltório e orientação à saída
CCR	Charnes, Cooper e Rhodes
CCR/E/I	CCR, modelo de envoltório e orientação à entrada
CCR/E/O	CCR, modelo de envoltório e orientação à saída
CRS	Constant Return to Scale
DEA	Data Envelopment Analysis
DMU	Data Making Unit
ISP	Internet Service Provider
MBF	Modelo Baseado nas Folgas
MPSS	Most Productive Scale Size
PL	Programação Linear
PPL	Problema de Programação Linear
RNC	Retorno Não-Crescente
RND	Retorno Não-Decrescente
SCOOP	Scientific Computation of Optimal Programs
VRS	Variable Return to Scale

Sumário

Capítulo 1	1
Introdução	1
1.1. Definição do Problema.....	1
1.2. Objetivos.....	3
1.2.1. Objetivo Geral da Dissertação.....	3
1.2.2. Objetivos Específicos.....	3
1.3. Relevância.....	4
1.5. Organização da dissertação.....	6
1.6. Comentários	7
Capítulo 2	8
Fundamentação Teórica.....	8
2.1. Programação Linear.....	8
2.1.1. Histórico.....	8
2.1.2. Fundamentos	9
2.1.3. Algoritmo Simplex.....	10
2.2. Breve Histórico da DEA	11
2.3. Conceitos da DEA	13
2.3.1. Folgas.....	15
2.4. Modelos da DEA.....	16
2.4.1. CCR (Charnes, Cooper e Rhodes)	17
2.4.2. BCC (Banker, Charnes e Cooper).....	19
2.4.3. Modelo Aditivo de Folgas ou Modelo Baseado nas Folgas (MBF).....	22
2.4.4 Outros modelos da DEA	23

2.5 Comentários	24
Capítulo 3	25
3.1. Cenário.....	25
3.1.1. Dados Reais	25
3.2. Metodologia	27
3.2.1. Arquitetura proposta	29
3.3. Normalização dos Dados	31
3.3 Dados normalizados na DEA com seus respectivos cálculos	32
3.4 Comentários	38
Capítulo 4	39
Resultados e Discussões	39
4.1. Ambiente de Desenvolvimento	39
4.2. Resultados Obtidos	40
4.2.1. Resultados das DMUs analisadas segundo o CCR	40
4.2.2 Resultados das DMUs analisadas segundo o BCC	44
4.2.3 Resultados das DMUs analisadas segundo o MBF	48
4.3. Análise dos resultados obtidos	48
4.4. Comentários	49
Capítulo 5	50
5.1 Conclusões	50
5.2 Trabalhos futuros.....	51
Referências Bibliográficas	53

Capítulo 1

Introdução

Este trabalho apresenta uma dissertação, submetida para obtenção do grau de Mestre, no Programa de Pós-Graduação em Ciências da Computação, do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará.

Este capítulo está dividido desta forma: 1.1) apresenta o problema e a motivação do trabalho; 1.2) apresenta o objetivo geral e os objetivos específicos da dissertação; 1.3) mostra a relevância científica do trabalho; 1.4) apresenta como a dissertação está organizada, e; 1.5) mostra os comentários finais acerca do capítulo.

1.1. Definição do Problema

A Internet vem ganhando novos adeptos a cada dia estimulando o surgimento de novos serviços e de seus prestadores de serviços, denominados ISPs (*Internet Service Providers*) ou Provedores de Internet. Os ISPs estão interligados à Internet, através de outros Provedores de Serviço de Interconexão, chamados *backbones*, geralmente empresas de telecomunicações.

A Internet foi criada, comercialmente no início da década de 90, nos Estados Unidos, com a migração de uma rede, meramente, acadêmica e com propósitos de disseminação de conhecimentos. A grande transformação de interesses foi gerada com a criação da WWW (*World Wide Web*), pelo cientista britânico Tim Berners-Lee, radicado no CERN, em Genebra na Suíça, em 1989, que propôs a criação de páginas navegáveis via hipertexto para facilitar as limitações de velocidade da época. A invenção de Berners-Lee gerou um grande facilitador de publicações de ideias na Internet e, então, assim cientistas de todo o mundo puderam ficar a par do estado da arte de várias linhas de pesquisa.

No Brasil, a Internet chegou na conferência ECO-92, mas só começou a ser explorada comercialmente, na metade da década de 90, pois antes só existiam conexões acadêmicas. A primeira conexão, no Brasil, foi em 1988, no LNCC (Laboratório Nacional de Computação Científica), que trabalhava com uma conexão de 9.600 bps (bits por segundo) interligada com a BITNET.

A exploração comercial da Internet, no Brasil, começou na metade da década de 90, chegando ao Nordeste do Brasil, em 1996, com o provedor chamado Openline, situado em João Pessoa-Paraíba. A conexão do cliente, via ISP com a Internet, era através de modems que

evoluíram com o tempo, passando a trabalhar na capacidade máxima do canal telefônico, que é de, aproximadamente, 56,6 kbps (kilo bits por segundo), segundo o teorema de Shannon.

O rápido crescimento da Internet exigiu que os *sites* das empresas e de usuários se tornassem dinâmicos, i.e., não fossem mais meramente ilustrativos, mas que pudessem interagir com outras tecnologias, incorporando linguagens de programação e Bancos de Dados aos simples hipertextos, visando à venda de produtos na Web. Esta transformação criou o comércio eletrônico ou e-commerce.

O desenvolvimento de novos serviços e de *sites* cada vez mais especializados, impulsionou o desenvolvimento de novas tecnologias de conexão dos ISPs para os clientes, objetivando o aumento da velocidade nas conexões de última milha. Entre estas tecnologias podem ser citadas: ADSL (*Asymmetric Digital Subscriber Line*), Internet via cabo, Internet via satélite, Internet sem fio para pequenas (WLAN – Wireless LAN), médias (WMAN – Wireless MAN, compatíveis com o protocolo 802.11) e longas distâncias (WiMax, HSDPA (*High Speed Downlink Packet Access*), etc.), fibra óptica, entre outras.

Estas tecnologias fizeram com que os *backbones* investissem muito em sua infraestrutura para oferecer serviços mais rápidos, mais baratos e mais confiáveis para os ISPs. Os provedores de conexão tiveram, também, de investir em capacitação e melhoria na sua infraestrutura para atender um número cada vez maior de clientes conectados.

O preço de ligação de um ISP com um *backbone* varia muito de uma região para outra, principalmente pelo fato de que, em regiões metropolitanas, onde o poder aquisitivo é maior, tem-se uma concorrência maior entre os *backbones*, o que diminui o preço da conexão. Assim, em regiões mais remotas, esta concorrência se dá entre duas ou, no máximo, três empresas de telecomunicações, o que faz com que o preço da conexão, nestas regiões, torne-se muito alto. O interior das regiões Norte, Nordeste e Centro-Oeste do Brasil fazem parte desta realidade apresentando preços elevados.

A diminuição do preço do *hardware* e a interiorização dos provedores de Internet no país fizeram com que os negócios se expandissem, sendo que muitos destes provedores visam mais o lucro do que a qualidade do serviço oferecido. Em horário comercial, quando o tráfego de dados é intenso, os serviços oferecidos por muitos ISPs não chegam a atingir 20% ou 30% da capacidade contratada e, desta forma, os clientes são penalizados pela contratação de sub-serviços.

Mesmo assim, muitas ferramentas de divisão da largura de banda entre os clientes são utilizadas tanto em software, como em hardware; bem como o uso de proxies, para mascarar a velocidade de conexão do provedor para o cliente, entre outras. Contudo, aplicações P2P (*Peer to Peer*) muitas vezes burlam essas ferramentas, enquanto os usuários chamados de *heavy users* (usuários pesados) ficam com a maior parte da velocidade, fazendo com que os demais usuários fiquem com fatias de velocidade menores. Logo, fatores preponderantes, para que um ISP possa oferecer bons serviços aos clientes são a contratação de um bom profissional de administração de redes, um profissional para projetar e montar uma boa infra-estrutura física da rede, contratação de um *link* que atenda as exigências contratuais de seus clientes, uma boa rede de suporte técnico e um administrador de empresas competente que consiga trabalhar, com todas estas variáveis citadas, trazendo lucros para empresa.

Este trabalho visa a aumentar o nível de satisfação dos clientes e das empresas, empregando uma técnica de estimação de uma fronteira Pareto-Koopman [KOOPMANS 1951] de pontos de referências (*benchmarks*) eficientes, baseados na Análise Envoltória de Dados (DEA – *Data Envelopment Analysis*), proposta originalmente por Charnes, Cooper e Rhodes em 1978 [COOPER e ZHU 2004]. Entretanto, esta mesma metodologia foi usada para estimar a eficiência dos ISPs, localizados no interior do estado do Ceará e Paraíba, tanto no ponto de vista da satisfação do cliente, quanto da maximização dos lucros das empresas.

1.2. Objetivos

1.2.1. Objetivo Geral da Dissertação

Avaliar as eficiências dos provedores de Internet com DEA, através de ferramentas computacionais desenvolvidas para este propósito, com o intuito de prover fronteiras eficientes Pareto-Koopman sob várias óticas, i.e., aumentar o nível de satisfação dos clientes e das empresas usadas para estimação das fronteiras.

1.2.2. Objetivos Específicos

- a) Estudar a DEA, seus modelos, formas, orientações e aplicabilidades;
- b) Investigar e identificar as principais ferramentas computacionais especializadas em DEA quer sejam livres ou proprietárias;
- c) Levantar os dados relevantes com os diversos ISPs que farão parte do modelo;
- d) Especificar e desenvolver um software para computar as fronteiras de eficiência, segundo o modelo, forma e orientação que se deseja;

- e) Apresentar os resultados, obtidos através da DEA e do software especialista desenvolvido, a respeito das empresas em questão;
- f) Mostrar a importância da abordagem DEA do ponto de vista econômico, além do ponto de vista tecnológico e especialista.

1.3. Relevância

Por ser a Internet uma realidade no cotidiano de pessoas e empresas ao redor do globo, muitos destes entes nortearam-na com uma responsabilidade descomunal, devido ao ganho de produtividade oferecido por ela. Assim, é de extrema importância prover serviços satisfatórios aos usuários que a utilizam na ponta; por conseguinte este serviço, em muitas ocasiões, é ofertado de maneira insatisfatória pelos prestadores de serviços denominados ISPs que, em grande parte, à medida que novos clientes aderem à utilização do serviço, tanto antigos, quanto novos clientes se depararão com uma situação cada vez mais corriqueira que é a diminuição na vazão ou velocidade contratada em detrimento de um lucro maior.

Por sua vez, a simples melhora no serviço, oferecido aos clientes pode desencadear um rápido processo falimentar dos ISPs, visto que estes têm de desembolsar uma quantidade fixa, mensalmente, para o pagamento do *link* à empresa de interconexão. Contudo, com a aplicação da Análise Envoltória de Dados (DEA), pode-se estimar uma fronteira de eficiência que pode contemplar as duas óticas com presteza, ou seja, tanto pode agradar o cliente com a maximização do desempenho por ele contratado, quanto pode buscar uma melhora relativa aos lucros dos ISPs.

Portanto, a solução do problema proposto é relevante, pois as firmas necessitam de melhor planejar seus processos produtivos, com a finalidade de perdurarem no mercado; além de, também, poderem projetar a entrada de novos clientes, de maneira que clientes antigos e novos saíam satisfeitos com os serviços por elas oferecidos.

1.4. Trabalhos relacionados

A Análise Envoltória de Dados (DEA) tem sido largamente utilizada, mais especificamente na dissertação de PhD. de G. Tavares, onde desenvolveu um apanhado bibliográfico da DEA, de 1978 a 2001 quando, até aquela data, existiam 3600 artigos escritos, além de diversos livros. Assim, mais de 1600 autores, em cerca de 42 países incluíam a avaliação de forças policiais, as junções econômicas e performances ambientais de empresas públicas na Europa, bem como o progresso social das sociedades árabes do Oriente Médio e Norte da África [FERREIRA e GOMES 2009].

Outra rica fonte a respeito da DEA é o sítio de Ali Emrouznejad, intitulado de DEAZONE. Tal sítio contém tutoriais, uma lista de 320 dissertações, várias publicações e livros a respeito da DEA.

Conforme citado, a DEA teve sua concepção no trabalho de dissertação de PhD. de Ewardo Lao Rhodes em 1978 [CHARNES, COOPER e RHODES 1978]. Em seu estudo, Rhodes avaliou o desempenho de um programa educacional denominado “*Follow Through*” (em português: Siga totalmente) e, neste mesmo trabalho, foram criados os termos DMUs (*Data Making Units* – Unidades de Tomada de Decisão), além das entradas ou insumos (*inputs*) e produtos ou saídas (*outputs*), sem esquecer, também, de toda uma base formal.

No Brasil a DEA vem recebendo, também, uma atenção crescente, destarte o trabalho de Villarroel et. al. (1999) que mostrou uma proposta para modelar indicadores de saúde das Microrregiões do Estado de Santa Catarina, usando DEA. O trabalho de MACEDO et. al. (2005) avaliou a eficiência dos investimentos em TI (Tecnologia da Informação) de empresas de tecnologia usando a DEA; Resende Neto (2006) trouxe uma análise de performance de fundos de investimento no país através da DEA.

Quintanilha e Ho (2006) publicaram um estudo sobre DEA para comparar a eficiência das ações de monitoramento de risco de incêndio, em municípios da Região Amazônica brasileira.

Chaves e Thomaz (2008) avaliaram o desempenho das agências da Previdência Social na região metropolitana de Fortaleza, Ceará, além de outras agências do interior do estado do Ceará, avaliando um número considerável de 22 DMUs. Tal estudo teve papel preponderante para melhoria no atendimento, destas agências, ao ser utilizada a DEA para gerar uma fronteira eficiente entre as agências analisadas, além de uma projeção de como as agências ineficientes poderiam acompanhar os seus parceiros de referência.

Existe uma gama de softwares que consegue resolver os cálculos das DMUs, através da DEA, usando a programação matemática (linear), dos quais podem ser citados: o MATLAB, que trabalha com programação matemática, centrado em operações, envolvendo vetores e matrizes; o LINDO, que resolve problemas de programação linear através do algoritmo Simplex; o LINGO, que resolve problemas, usando a programação linear, não-linear e inteira. Dentre estes podemos, ainda, citar o projeto nacional, desenvolvido por professores e alunos da Universidade Federal de Viçosa-MG, denominado PROLIN, que resolve sistemas através da programação linear na Web.

Por outro lado existe, também, uma série de softwares, especialistas na DEA, sendo que vários destes livres, podem ser citados: o DEAP, os nacionais SIAD, IDEAL que é o único dos softwares a mostrar uma fronteira tridimensional e o SAED. Explorando a DEA, comercialmente, existem vários softwares, tais como: o DEA Excel Solver, o PIMSoft, o Frontier Analyst, dentre outros. Um dos principais softwares gratuitos pode ser citado: o DEA-Excel que consegue trabalhar com até duzentas DMUs e vinte entradas e saídas.

Várias ferramentas foram propostas, a fim de se checar a vazão fim a fim entre hosts que compõem uma rede, em especial a Internet. Estas ferramentas foram desenvolvidas baseadas em técnicas de medições passiva e ativa [BARFORD e SOMMERS 2004]. Assim, a monitoração de tráfego é dada, através do registro passivo de pacotes em um enlace, enquanto as medições ativas de desempenho usam o envio de pacotes, denominadas de sondas, de medição [ZIVIANI e DUARTE 2005].

Entretanto, estas técnicas levam em conta, apenas, o lado do desempenho da rede de computadores, mas não avaliam o desempenho econômico. Por conseguinte, os clientes dos ISPs podem recorrer a ferramentas Web para conseguirem analisar a velocidade com a qual estão conectados, em um dado momento e, com a finalidade de saber se os serviços, por eles contratado, estão em conformidade. Entre estas ferramentas Web podem ser citadas: o velocímetro do ISP RJNET, o velocímetro da Beltrônica, etc.

O clássico trabalho de Jain (1991) relaciona as três principais técnicas, aplicadas à análise de desempenho em redes de computadores. São elas: a simulação, a modelagem analítica e a experimentação.

1.5. Organização da dissertação

Esta dissertação está organizada em 5 (cinco) Capítulos e suas respectivas referências bibliográficas.

O primeiro Capítulo é uma introdução à dissertação, em que mostra uma contextualização do problema, a motivação, os objetivos geral e específicos a serem alcançados, além da relevância do problema e as suas contribuições. Tal capítulo remete-se a uma revisão bibliográfica do assunto abordado, apresentando a DEA, sendo usado para estimação de fronteiras eficientes em várias áreas distintas, além de mostrar a estrutura da dissertação.

O segundo Capítulo apresenta os conceitos, as técnicas empregadas e as fundamentações teóricas usadas neste trabalho. Neste capítulo é apresentada a programação

linear, com seu histórico e fundamentos, além de abordar o algoritmo Simplex. Também é formalizado o problema de satisfação dos clientes, versus a maximização dos lucros, pelas DMUs (*Data Making Units*) ou Unidades de Tomada de Decisão em estudo. No caso, os ISPs. É apresentada, também, a essência da DEA desde a sua concepção. Neste, são conhecidos os principais modelos da DEA, com suas respectivas formas e orientações, bem como mostra outros modelos DEA.

O terceiro capítulo é devotado ao cenário, utilizado na pesquisa e na metodologia usada para o desenvolvimento deste trabalho, acompanhado pela arquitetura proposta, do fluxograma algorítmico utilizado e mostrando o processo de normalização dos dados, passo este importante para se alcançar uma maior veracidade nos cálculos DEA.

No quarto capítulo serão mostrados os resultados, obtidos com o uso do *Solver* DEA proposto, além de outros softwares auxiliares, para se obterem resultados mais apurados e, também, para servirem como provas dos resultados alcançados pelo *Solver* DEA proposto.

Finalmente, o quinto capítulo encerra a dissertação com as conclusões do trabalho realizado e as seguidas das respectivas futuras pesquisas.

1.6. Comentários

Este capítulo mostrou o problema que se quer resolver, os objetivos que devem ser alcançados ao término do trabalho dissertativo, a relevância do problema para os ISPs e outras empresas em geral, além de expor a estrutura da dissertação.

Ainda foi apresentada a Análise Envoltória de Dados (DEA) que tem se tornado pertinente, ao analisar o desempenho quantitativo de empresas ou firmas ou Unidades Tomadores de Decisão (DMUs), acerca das últimas três décadas. Além disso, o problema de análise de desempenho de redes de computadores é bastante conhecido na literatura e empregado para prover maior eficácia no transporte fim a fim de pacotes inter-redes. Assim, este capítulo foi remetido às principais pesquisas e ferramentas, usadas nestas áreas, nos últimos anos.

Capítulo 2

Fundamentação Teórica

Este capítulo abordará os aspectos teóricos que fundamentam a proposta defendida neste trabalho. Na primeira parte do capítulo, serão apresentados os conceitos e fundamentos da programação linear, passando pelo algoritmo Simplex. Na segunda parte, será mostrado um histórico da DEA. A terceira parte tratará de alguns conceitos usados pela DEA. A quarta parte apresentará os formalismos da DEA, direcionada, principalmente, para os modelos DEA, usados neste trabalho; bem como serão apresentados alguns modelos adicionais.

2.1. Programação Linear

Nesta seção será mostrado um apanhado da programação linear e do algoritmo Simplex, usado para solucionar PPL's.

2.1.1. Histórico

A programação linear é baseada na álgebra linear. Assim são usados cálculos, iterativamente, para solucionar problemas através de algoritmos computacionais. Problemas de Programação Linear (PPL) são problemas de otimização, sendo que a função objetivo e as suas restrições são lineares.

De acordo com Lins e Calôba (2006), dois trabalhos científicos influenciaram o desenvolvimento da Programação Linear (PL). Foram eles:

1. Desenvolvimento da Teoria dos Jogos por John Von Neumann em 1928;
2. Criação da Análise insumo-produto por Leontief em 1936.

Outros trabalhos científicos surgiram neste período, sendo formulados sob a forma de restrições, aumentando o interesse da comunidade científica para solucionar estes problemas. Daí, surgiram os PPL, tais como: o problema da organização e planejamento de produção, o problema do transporte, o problema da dieta, entre outros.

A proposição e a consequente resolução destes problemas impulsionou a consolidação de uma nova área de pesquisa, denominada de Pesquisa Operacional, que usa a PL para solucionar problemas de otimização.

As luzes da pesquisa operacional começaram a clarear o mundo científico na 2ª Guerra Mundial nos EUA. Nesta época, foi criado um projeto chamado de SCOOP (*Scientific*

Computation of Optimal Programs) que avaliava a viabilidade da aplicação de métodos matemáticos em problemas orçamentários e de planejamento militar. Um dos membros do SCOOP era George Dantzig, que no mesmo ano que se iniciou o projeto, 1947, juntamente com sua equipe de desenvolvimento, desenvolveram um método de solução denominado Simplex.

Com o passar do tempo, o aumento do interesse pela PL cresceu entre os cientistas, bem como pelas áreas dos PPL. Assim, podem ser encontradas soluções com PL para problemas de meio ambiente, transporte, energia, problemas sociais, industriais e em outras áreas.

Outro marco importante na PL foi o desenvolvimento do algoritmo dos pontos interiores por Karmakar (1984). O algoritmo dos pontos interiores era um algoritmo polinomial, para resolução de PPL's, que obtém resultados com eficiência e eficácia. O objetivo do algoritmo dos pontos interiores é transformar as restrições de desigualdade em restrições de igualdade com a incorporação de variáveis de folga e do uso de uma barreira logarítmica à função objetivo, para evitar a negatividade das variáveis de folga, fazendo com que a solução esteja na região de soluções viáveis.

Atualmente, a PL vem auxiliando outras ciências em busca de se otimizar os resultados através da solução de vários PPL's. Uma das técnicas que usam a PL é a DEA, que se utiliza do Simplex para estimar sua fronteira de eficiência.

2.1.2. Fundamentos

Segundo Lins e Calôba (2006), o uso de modelos de PL compreende as etapas de formulação do PPL, resolução e validação dos resultados. A primeira parte é a representação matemática do problema proposto e, nesta etapa, serão definidas as variáveis e suas relações, reconhecidas as restrições relevantes e expresso(s) o(s) objetivo(s).

Assim, para a resolução de PPL's, são necessários os seguintes elementos:

- Variáveis de decisão: são relevantes ao problema, passíveis de quantificação e disponíveis;
- Função Objetivo: é uma função, produto de coeficientes pelas variáveis de decisão, ao qual o problema deseja otimizar, ora através da maximização, ou da minimização;
- Restrições: são os elementos restritivos que todo problema possui, tal como restrições de capital para investimento ou de escassez de recursos.

2.1.3. Algoritmo Simplex

O Algoritmo Simplex foi proposto por George Dantzig (1963), em 1947, sendo desenvolvido pela equipe do SCOOP. O nome do algoritmo veio do conceito de um simplex, i.e., uma figura geométrica de $N+1$ vértices em N dimensões, i.e, um segmento de reta sobre uma reta, ou um triângulo sobre um plano.

O algoritmo Simplex busca as chamadas soluções básicas viáveis, que são aquelas que se situam nos vértices da região limitadora de soluções ótimas. Assim, se o PPL tem solução ótima, haverá uma solução básica viável que será uma solução ótima [LINS e CALÔBA 2006].

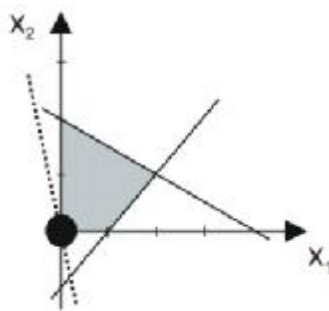


Figura 2.1. Espaço de busca de soluções do Simplex.

A sequência de passos lógicos finitos, usados pelo Simplex, são [LINS e CALÔBA 2006]:

- Buscar a primeira solução básica viável;
- Checar se existem vértices adjacentes viáveis com possibilidade de melhoria da função objetivo, isto é, executar um teste de otimalidade;
- Movimentar a solução para um vértice melhor, da seguinte maneira:
 - o Examinar qual dos $m-1$ vértices adjacentes otimiza melhor a função objetivo, sendo que m é o número de restrições do problema;
 - o Deter o acréscimo da variável que entrou na base do limite da restrição (i.e, o vértice);
 - o Solucionar o problema para este vértice com o pivoteamento das equações;
- Aplicar o teste de otimalidade;
 - o Se a solução for ótima, parar a execução do algoritmo;

- Se a solução não for ótima, realizar uma nova iteração;
- Iterar o algoritmo até encerrar o problema.

O algoritmo Simplex se dá bem na maioria dos problemas propostos, entretanto um PPL pode admitir vários tipos de soluções, a saber:

- Uma única solução ótima;
- Infinitas soluções ótimas;
- Problema inviável;
- Problema ilimitado.

2.2. Breve Histórico da DEA

A DEA se confunde com a Teoria da Produção, pois ambas buscam a eficiência das empresas que são submetidas às análises. A DEA surgiu de conceitos da produção microeconômica que, segundo Ferreira e Gomes (2009), foram registrados no livro de Johann-Heinrich Von Thünen (1783-1850), intitulado de “O Estado Isolado” e, tal obra, tratava de melhoria na produção agrícola.

Jon Von Neumann, notável cientista com contribuições em muitas áreas, em seu trabalho [NEUMMAN 1945], apresentou um modelo geral de equilíbrio econômico, mostrando que os bens são produzidos por si, numa espécie de processo circular, e que o número de processos técnicos de produção pode ser maior do que o de mercadorias. Este trabalho usou a ideia de retornos constantes. Ele dizia, também, que as quantidades usadas e as produzidas dos bens no qual os processos de produção dependem devem ser, obviamente, não negativas.

Lins e Calôba (2006) apontam que, em 1951, Debreu introduziu uma medida radial de eficiência técnica, denominado de coeficiente de utilização de recursos. Tal medida radial pretende alcançar a máxima redução equiproporcional de todas as entradas ou a máxima expansão equiproporcional de todas as saídas; sendo tanto as entradas, como as saídas, independentes de unidades. O grande problema deste modelo é que o mesmo pode ser ineficiente na definição de Koopman-Pareto.

Outro trabalho seminal foi o de Dantzig (1963) que influenciou um ramo da ciência em especial, denominado Pesquisa Operacional, muito importante para obter otimização dos processos produtivos. Dantzig mostrou os fundamentos do método Simplex, que através de iterações no espaço de soluções; em especial, nos vértices, busca uma solução possível se ela

existir. O Simplex trabalha com a introdução de variáveis artificiais, resultando num problema auxiliar na forma canônica, daí o Simplex é empregado. Este método consiste em uma sequência de operações pivôs, referenciadas como Fase I que resultarão em uma sucessão de formas canônicas, a fim de encontrar uma solução possível num espaço de soluções. Caso a forma canônica produza cada solução, então o Simplex é novamente aplicado em uma segunda sucessão de operações de pivôs chamada de Fase II. Caso a solução não encontre uma solução o algoritmo terminará em um número finito de iterações.

Outro conceito empregado é da fronteira de eficiência, baseada no trabalho de Koopmans (1951), que foi fortemente influenciado pelo trabalho do economista, sociólogo e engenheiro ítalo-suíço Vilfredo Pareto. Por isso a menção a fronteira de otimalidade de Pareto-Koopmans, também conhecida como fronteira ótima de Pareto, é baseada no pensamento abaixo:

“Situação hipotética em que ninguém pode alterar sua situação social, buscando uma posição mais cômoda, sem que isso provoque uma mudança de outra pessoa para uma posição que não lhe agrade”.

Como, novamente, exposto por Ferreira e Gomes (2009), mostrou-se, neste trabalho, a grande importância do cientista James Farrel em seu trabalho: “The Measurement of Productive Efficiency” em 1957, onde Farrel desenvolveu métodos para avaliar a produtividade, através dos conceitos de análise de atividades; mas tal trabalho limitou-se, usando apenas um único produto, pois, ao se trabalhar com múltiplos produtos, o modelo tornava-se ineficiente para a aplicação de um grande conjunto de dados. Tal trabalho lhe valeu o prêmio Nobel de economia.

A DEA apareceu como método de estimação de fronteira eficiente, a partir da década de 70, no esforço de Edwardo Lao Rhodes em obter o grau de PhD [COOPER et. al. 2007]. Deste modo, em 1978, o termo DEA apareceu em um relatório chamado de: “*A Data Envelopment Analysis Approach to Evaluation of the Program Follow Through Experiment in U.S. Public School Education*”. Tal programa avaliava a eficiência de escolas públicas que pertenciam ao programa educacional *Follow Through* para alunos carentes, tornando-se um marco, para aplicar princípios de projetos estatísticos, a fim de mensurar um conjunto de escolas que faziam parte deste projeto. Tal trabalho introduziu o termo DMU (*Data Making Unit*), onde, naquele caso, eram as escolas avaliadas. Por ter utilizado um grande conjunto de dados, o trabalho de Rhodes não teve nenhum problema com as variáveis de entrada e saídas usadas. Assim, a DEA corrigiu os erros oriundos do trabalho de Farrel.

2.3. Conceitos da DEA

Por ser um modelo de estimação de fronteira de produção eficiente, a DEA baseia-se na premissa básica da produção, onde um conjunto de insumos unidos serão transformados através de processos produtivos, com o intuito de gerar produtos (saídas), para obter o maior lucro possível. Sabe-se que, nem sempre, esta produção trará lucros para as empresas que os manufaturam, devido a diversos fatores.

As empresas, ao se utilizarem dos processos produtivos empregados, podem ser eficientes se atingirem os objetivos propostos, sem levar em conta os recursos utilizados; ou seja, tais empresas podem ser eficientes, mas não lucrativas. Outro conceito é de produtividade, que está diretamente ligado ao modo de como os recursos de uma empresa estão sendo empregados, com a finalidade de obter um produto. Este conceito de produtividade se dá pela relação entre a produção com o insumo usado:

$$\frac{\text{Produção}}{\text{Insumo}} \quad (1)$$

A DEA, diferentemente do método de Farrel, não se limita a um único insumo e a um único produto, usando, assim, um número qualquer de entradas (insumos) e saídas (produtos) com o intuito de avaliar as empresas do ponto de vista da eficiência no emprego das entradas e saídas. O objetivo da DEA é fazer com que as DMUs possam minimizar a quantidade de entradas, para se obter a maximização das saídas, objetivando um maior lucro para as firmas avaliadas, além de definir quais das empresas avaliadas estão trabalhando com maior produtividade, pois, desta forma, a DEA poderá calcular como as empresas mal avaliadas poderiam trabalhar como as outras que são suas parceiras de referência (*benchmarks*).

Outro conceito de suma importância é o de eficiência técnica que compara o que foi produzido por uma unidade com o que poderia ser produzido [FERREIRA e GOMES 2009], da seguinte forma:

$$\frac{\text{Produção}}{\text{Insumo}} \text{ realizada, comparada com } \frac{\text{Produção}}{\text{Insumo}} \text{ a mais adequada}$$

Em relação aos dados, utilizados pela DEA, na entrada ou na saída, não se têm nenhuma restrição referentes às medidas adotados, podendo estes ser uma mescla entre preços, vazão de conexões fim a fim, funcionário num dado setor, quantidade de peso em quilogramas, entre outros.

Por outro lado as DMUs não necessitam ser, exclusivamente, empresas ou corporações, mas podem ser setores de empresas e/ou órgãos governamentais, filiais, escolas, órgãos governamentais, etc. Assim, a DEA pode ser usada, sob qualquer ótica, para avaliar um conjunto de entes produtivos, através de suas variáveis de entrada e de saída.

Pode-se afirmar que a DEA é um método matemático não paramétrico, pois não se utiliza de inferências estatísticas, nem se apega a medidas de tendência central, testes de coeficientes ou formalizações de análise de regressões [FERREIRA e GOMES 2009]. A ideia central da DEA, segundo COOPER et. al. (2007) é a avaliação de performance de atividades ou organizações, através da análise de eficiência de unidades, centrados em conceitos como produtividade e eficiência técnica. DEA usa técnicas como a programação matemática, que trata um grande número de variáveis e restrições. Isso relaxa os requisitos que são, frequentemente, encontrados, quando se está limitado à escolha de poucas entradas e saídas, porque as técnicas empregadas encontram dificuldades.

Depois que uma fronteira linear de eficiência, dividida por partes (*linear piecewise frontier*), com características convexas, for gerada, serão visualizadas DMUs que estão nesta fronteira e outras que estão abaixo da fronteira e que têm de alcançar seus parceiros de referência (*benchmarks*); ou através da minimização das entradas, sem a diminuição das saídas, ou a maximização dos lucros (saídas), sem aumentar as entradas, caracterizando o chamado movimento radial.

Na figura 2.2 é apresentado o modelo de envoltório de dados, baseado nas saídas produzidas por cada empresa. Deste modo, as posições representadas pelas empresas 1, 2 e 4 mostram que estas estão formando uma fronteira de eficiência e, assim, a empresa 3 está abaixo desta fronteira, precisando aumentar alguma(s) medida(s) para alcançar o nível de performance das empresas concorrentes. A figura a seguir ainda mostra que a empresa 2 é o parceiro de referência (*benchmark*) da empresa 3.

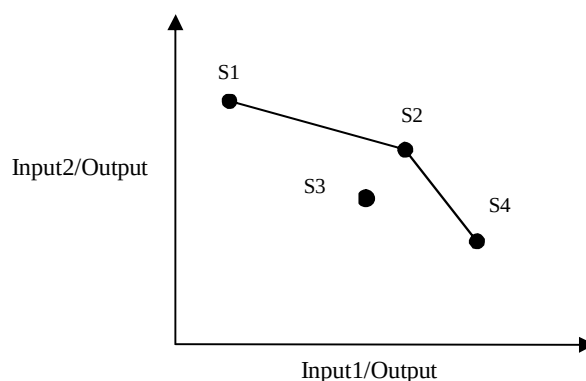


Figura 2.2. Envoltório de Dados

2.3.1. Folgas

A forma da fronteira não-paramétrica linear, na DEA, pode causar dificuldades na medida de eficiência [COELLI 1996]. Este problema é gerado, devido às seções (pedaços ou vizinhos ou *peers*), usadas na fronteira linear paralelas aos eixos, o que não ocorrem em muitas funções paramétricas.

No método de Farrel, as medidas de eficiência técnica entre duas DMUs A e B são, respectivamente, OA'/OA e OB'/OB . Mas, como se pode observar, na figura a seguir, deve-se questionar se o ponto A' é eficiente, desde que se poderia diminuir mais a quantidade da entrada x_2 (usada pela quantidade CA'), para produzir a mesma saída. Este conceito é chamado de folga que, tanto pode ser de entrada, sendo chamada de excesso; como de saída, que é chamada de insuficiência de saída ou falta.

Na figura 2.3, a folga de entrada (excesso), associada ao ponto A' , é CA' da entrada x_2 . Quando existem mais do que uma entrada e/ou saídas, o cálculo das folgas não é uma tarefa simples. Adiante, serão mostrados como podem ser calculadas estas folgas nas entradas e nas saídas.

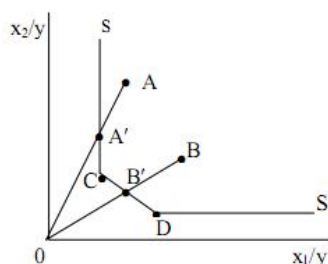


Figura 2.3. Medidas de eficiência e folgas de entrada [COELLI 1996].

2.4. Modelos da DEA

DEA, por ser um método eficiente para mensurar a eficiência das corporações, pressupõe tal fronteira sob duas óticas: i) diminuindo a quantidade de insumos, para produzir o mesmo produto, e; ii) aumentando a quantidade de produtos, mantendo a mesma proporção de insumos. Todas essas suposições são baseadas na ideia da Fronteira de Pareto.

A notação da DEA dispõe que os dados estão dispostos em K entradas e M saídas em cada uma das N firmas ou DMUs. A matrix KxN é chamada de matriz de entradas, sendo representada por X, e a matriz MxN é chamada de matriz de saídas que, sendo representada por Y, formam os dados de todas as N DMUs [COELLI 1996].

Uma das formas iniciais da DEA era através da forma fracionária, assim, para cada DMU deve-se obter uma medida, através de todas as saídas e sobre todas as entradas, tal como:

$$u' y_i / v' x_i \quad (2)$$

Onde u é um vetor Mx1 de saídas ponderadas (ou pesos), e v é um vetor Kx1 de entradas ponderadas. Tal suposição resultou em um modelo de programação matemática fracionária, para selecionar pesos ótimos de u e de v das DMUs, que pode ser solucionado através das fórmulas a seguir:

$$\max_{u,v} (u' y_i / v' x_i) \quad (3)$$

s.a.

$$u' y_j / v' x_j \leq 1, j = 1, 2, \dots, N \quad (4)$$

$$u, v \geq 0 \quad (5)$$

O problema da programação matemática fracionária é que ele tem um número infinito de soluções. Tal dificuldade foi contornada, através do uso da programação linear, mais especificamente pelo método Simplex, que resultou na formulação original da DEA, denominada de modelo CCR, uma alusão às iniciais dos mentores Charnes, Cooper e Rhodes.

Um outro problema encontrado da programação matemática fracionária, é que ela não garante que a restrição de não negatividade (5) gerará resultados positivos.

Deve ser lembrado que a razão entre as saídas virtuais pelas entradas virtuais não pode exceder o valor da unidade, ou seja, os valores têm de ser menores ou iguais a 1 (≤ 1).

Nas próximas seções serão mostradas as principais orientações da DEA, que são os modelos CCR e BCC (Barnes, Charnes e Cooper), além do modelo aditivo de folgas e, finalmente, serão mostrados alguns modelos complementares.

2.4.1. CCR (Charnes, Cooper e Rhodes)

O modelo CCR foi o modelo proposto, originalmente em 1978, pelos cientistas Charnes, Cooper e Rhodes, por isso esta orientação tem as iniciais destes. Este modelo pressupõe Retornos Constantes de Escala (RCE) ou CRS (*Constant Return to Scale*), i.e., que as entradas e as saídas são proporcionais entre si.

Enzo Mariano et. al. (2006) evidenciaram que o CCR desconsidera ganhos e escala quando calcula a eficiência. Portanto, a eficiência relativa de uma DMU é obtida através da divisão entre a sua produtividade e maior produtividade dentre as DMUs da observação. Assim, o formato da fronteira do modelo de eficiência CCR é uma reta que forma um ângulo de 45°.

Pelos problemas que foram mostrados da programação matemática fracionária, foi proposto o modelo de programação linear primal, através do método Simplex, para resolver problemas de otimização, obtido com a combinação linear das n variáveis de entrada e saída.

Na forma primal existem m restrições e estas restrições encontram-se em um limite superior (*upper bound*), em uma combinação linear de n variáveis. Na forma dual, estas mesmas restrições encontram-se em um limite inferior (*lower bound*). Todavia, ambas as formas geram, como solução, um vetor de s valores que alcançam os valores da função objetivo.

Assim, foi proposto o modelo CCR que transformou o problema de programação matemática fracionária num problema de programação matemática linear. Contudo, tal modelo possui tanto a forma primal, que também é chamada de modelo dos multiplicadores, como a forma dual, chamada de modelo de envoltório ou envoltório. Por isso, o nome Análise Envoltória de Dados.

Outro ponto a ser tocado é que todos os modelos, na DEA, possuem orientações para entrada e para saída, sendo que, em ambas as orientações, bem como nas formas dos multiplicadores e de envoltórios, são gerados valores idênticos. A única diferença considerável se dá em termos computacionais, pois o modelo de envoltório possui menos restrições do que o modelo dos multiplicadores, i.e., no modelo dos multiplicadores existem $N+1$ restrições, enquanto que no modelo de envoltório existem, $K+M$ restrições.

Abaixo, serão mostradas as fórmulas da DEA para o CCR, tanto para o modelo dos multiplicadores, quanto para o modelo de envoltório, seguindo ambas as direções (entradas e saídas). São elas:

Modelo dos Multiplicadores – Primal

Orientação a Insumo:

$$\text{Maximizar Eficiência} = \sum_{j=1}^m m_j y_{jo} \quad (6)$$

Sujeito a:

$$\sum_{i=1}^r v_i x_{io} = 1 \quad (7)$$

$$\sum_{j=1}^m m_j y_{jk} - \sum_{i=1}^r v_i x_{ik} \leq 0 \quad (8)$$

$$m_j, v_i \geq 0 \quad (9)$$

Onde v e μ são matrizes de entradas e saídas respectivamente. Vale ser ressaltado que tanto os modelos Primais, quanto Duais, em ambas as orientações, produzirão o mesmo resultado no modelo CCR.

Orientação a Produto:

$$\text{Minimizar Eficiência} = \sum_{i=1}^r v_i x_{io} \quad (10)$$

Sujeito a:

$$\sum_{j=1}^s m_j y_{jk} = 1 \quad (11)$$

$$\sum_{j=1}^s m_j y_{jk} - \sum_{i=1}^r v_i x_{ik} \leq 0 \quad (12)$$

$$m_j, v_i \geq 0$$

Modelo dos Envoltórios – Dual

Orientação a Insumo:

$$\text{Minimizar Eficiência} = q \quad (13)$$

Sujeito a:

$$qx_{io} - \sum_{k=1}^n l_k x_{ik} \geq 0 \quad (14)$$

$$\sum_{k=1}^n l_k y_{mk} - y_{mo} \geq 0 \quad (15)$$

$$l_k \geq 0 \quad (16)$$

Orientação a Produto:

$$\text{Maximizar Eficiência} = f \quad (17)$$

Sujeito a:

$$x_{io} - \sum_{k=1}^n x_{ik} l_k \geq 0 \quad (18)$$

$$\sum_{k=1}^n y_{mk} l_k - f y_{mo} \geq 0 \quad (19)$$

$$l_k \geq 0$$

Onde a eficiência θ será obtida da razão $1/f$. Nota-se, também, que todas as restrições não excedem a unidade, nem são menores que zero, caracterizando, assim, as porcentagens referentes às eficiências técnicas de cada uma das DMUs em questão.

Outra informação que poderá ser avistada nos vários livros, artigos e outros da DEA é que, geralmente, é utilizado o formato Modelo/Forma/Orientação para simplificar a descrição das fórmulas, i.e., ao usar o CCR na forma dos multiplicadores e, com a orientação para os produtos ou saída, a descrição das fórmulas seria CCR/M/O, onde o I é relativo às entradas ou insumos e o O é relativo às saídas ou produtos. No caso, quando se usa o modelo BCC, troca-se o CCR por BCC. Se for usada a forma de envoltório, é trocada a letra M pela letra E; entretanto, se for usada a orientação de entrada em detrimento da de saída, usa-se o I em vez do O.

2.4.2. BCC (Banker, Charnes e Cooper)

A suposição de Retornos Variáveis de Escala (RVE) ou VRS (*Variable Return to Scale*) só é apropriada quando todas as DMUs estão operando em uma escala ótima. A competição imperfeita, restrições de capital, regulamentações governamentais, etc. podem fazer com que as firmas não operem em uma escala ótima [COELLI 1996]. Através desta máxima, Banker, Charnes e Cooper sugeriram, em 1984, um ajuste ao modelo CCR. Desta forma, tal modelo ficou, também, conhecido por BCC, que são as iniciais dos criadores do modelo. Esse modelo pressupõe RVE; ou seja, pode possuir, de acordo com Enzo Mariano et. al. (2006), os três tipos de retorno de escala:

- **Retorno Crescente (RNC)** ou **Não-Decrescente** – quando o aumento do número de entradas ocasiona o aumento desproporcional maior no número de saídas, o que ocorre quando uma firma está operando muito abaixo de sua capacidade ótima.
- **Retorno Constante** – quando o aumento do número de entradas causa um aumento proporcional nas saídas, ou seja, uma firma com retorno constante, operando em sua capacidade ótima.
- **Retorno Decrescente (RND)** ou **Não-Crescente** - quando o aumento do número de entradas, ocasiona o aumento desproporcional menor no número de saídas; o que ocorre, quando uma firma está operando muito acima de sua capacidade ótima.

Com a presença de vários tipos de retorno, as restrições impostas aos modelos de multiplicadores e de envoltório mudam. Assim, quando é RCE $\mu_o=0$, quando RVE μ_o é livre, quando o RNC $\mu_o \leq 0$ e quando o RND $\mu_o \geq 0$.

Em concomitância ao modelo CCR, o modelo BCC também possui as formas: fracionária, dos Multiplicadores e de Envoltório. Outra característica marcante do modelo BCC é que este geralmente apresenta valores maiores que os do modelo CCR. Com isso, surge o conceito de eficiência de escala [FERREIRA e GOMES 2009], que resulta do nível de eficiência máxima mais adequada, em razão da tecnologia a ser adotada. Tal situação é chamada de Tamanho de Escala Mais Produtiva ou MPSS (*Most Productive Scale Size*). A eficiência de escala é calculada pela fórmula:

$$EFE = \frac{qCCR}{qBCC} \quad (20)$$

Outra diferença significativa do BCC, em relação ao CCR, é que a fronteira de eficiência, gerada pelo modelo BCC, é convexa, curva e não-radial. Para uma ilustração mais clara, observe a figura a seguir que pressupõe uma fronteira pontilhada linear, do modelo CCR, e uma fronteira convexa, curva e não radial, referente ao modelo BCC.

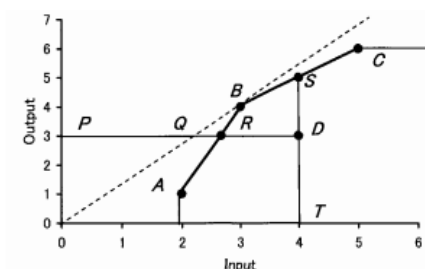


Figura 2.4. Fronteiras CCR x BCC [COOPER et. al. 2007].

As fórmulas para cálculo de eficiência técnica do modelo BCC, na forma Primal (Multiplicadores) e Dual (Envoltório), são apresentadas a seguir.

Modelo dos Multiplicadores – Primal

Orientação a Insumo:

$$\text{Maximizar a Eficiência} = \sum_{j=1}^m m_j y_{io} + m_o \quad (21)$$

Sujeito a:

$$\sum_{i=1}^j v_i x_{io} = 1 \quad (22)$$

$$\sum_{j=1}^m m_j l_{jk} - \sum_{i=1}^t v_i x_{ik} + m_o \leq 0 \quad (23)$$

$$m_j, v_i \geq 0$$

Orientação a Produto:

$$\text{Minimizar Eficiência} = \sum_{i=1}^r v_i x_{io} + v_o \quad (24)$$

Sujeito a:

$$\sum_{j=1}^s m_j l_{jo} = 1 \quad (25)$$

$$\sum_{j=1}^s m_j l_{jk} - \sum_{i=1}^r v_i x_{ik} + v_o \leq 0 \quad (26)$$

$$m_j, v_i \geq 0$$

Modelo dos Envoltórios – Dual

Orientação a Insumo:

$$\text{Minimizar Eficiência} = q$$

Sujeito a:

$$q x_{io} - \sum_{k=1}^n l_k x_{ik} \geq 0 \quad (27)$$

$$\sum_{k=1}^n l_k y_{mk} - y_{mo} \geq 0 \quad (28)$$

$$l_k \geq 0 \quad (29)$$

Orientação a Produto:

$$\text{Minimizar Eficiência} = f \quad (30)$$

Sujeito a:

$$x_{io} - \sum_{k=1}^n x_{ik} l_k \geq 0 \quad (31)$$

$$\sum_{k=1}^n y_{mk} l_k - f y_{mo} \geq 0 \quad (32)$$

$$l_k \geq 0$$

Como no CCR, o BCC produzirá resultados idênticos, nos modelos Primais e Duais, em ambas as orientações. Além disso, algumas destas fórmulas serão preenchidas com os dados das DMUs reais, utilizadas neste trabalho, para que se possa calcular a eficiência técnica relativa à DMU num software de resolução de problemas de programação linear chamado LINDO.

2.4.3. Modelo Aditivo de Folgas ou Modelo Baseado nas Folgas (MBF)

O Modelo Aditivo de Folgas foi proposto por Charnes et. al. (1985) e, também, é conhecido como Medida de Ajuste de Escala ou RAM (Range-Adjusted Measure), ou Modelo Baseado nas Folgas (MBF). Este modelo é importante, pois segundo Ferreira e Gomes (2009) ele considera, ao mesmo tempo, tanto a possibilidade de redução dos insumos, quanto o aumento da produção, mas baseado nas folgas dos insumos e produtos.

O MBF inclui folgas diferentes de zero que dão uma estimativa de excessos de entrada e deficiências nas saídas que poderiam ser melhoradas sem a perda de qualquer entrada ou saída [LEVERTY e GRACE 2007].

Novamente o trabalho de Ferreira e Gomes (2009) mostra que o MBF pressupõe que o valor marginal das folgas das entradas e saídas maiores que zero seja igual. Assim, deve-se ter cuidado ao empregar tal modelo, pois: a) as unidades de medida, usadas para as entradas e saídas influenciam nos resultados, i. e., unidades de medida diferentes resultam em medidas de eficiência diferentes; b) é necessário ter cuidado ao somar medidas que não sejam comensuráveis. Portanto, é muito recomendado que, quando usar a DEA, tenha-se bastante experiência com o setor ou organização em análise.

Além de todas estas suposições, o MBF se dá ao luxo de introduzir pesos nas unidades de entrada saída para agir no modelo, de acordo com o julgamento de especialistas. Estes pesos possuem o viés da pessoalidade, ou seja; para um especialista uma unidade pode possuir um determinado peso, mas para outro especialista já pode ser outro peso.

O MBF é importante, pois uma empresa pode ter eficiência técnica de 100%, mas mesmo assim, pode possuir excessos de insumos ou falta de produtos. Essas empresas são chamadas de falso eficientes. Porém, tal modelo ajuda à(s) empresa(s) analisada(s) a se tornar(em) realmente eficiente(s).

Diferentemente dos modelos CCR e BCC, o MBF só possui as formas de envoltório, sem nenhuma orientação, como pode ser visto nas fórmulas abaixo.

Modelo Aditivo ou Modelo Baseado em Folgas

$$\text{Maximizar } \sum_{i=1}^r s_i^- + \sum_{m=1}^s s_m^+ \quad (33)$$

$$(l, s_{ik}^-, s_{mk}^+)$$

Sujeito a:

$$\sum_{k=1}^n l_k x_{ik} + s_i^- = q x_{io} \quad (34)$$

$$\sum_{k=1}^n l_k y_{mk} - s_m^+ = y_{mo} \quad (35)$$

$$s_i^-, s_m^+, l_k \geq 0 \quad (36)$$

2.4.4 Outros modelos da DEA

A DEA vem despertado o interesse de vários pesquisadores no mundo inteiro, por isso vários destes proporam novos modelos para o DEA. Abaixo, estão listados alguns destes modelos com uma breve descrição:

- **Modelo de Medidas Específicas (MME):** variáveis não discricionárias, não controláveis, exógenas ou fixas. Este modelo trabalha com variáveis que estão fora do controle dos tomadores de decisão, e são chamadas de variáveis discricionárias, exógenas ou fixas,

em que os tomadores de decisão não podem manipular seus valores. Desta forma, o MME leva em conta tanto as variáveis discricionárias, quanto às não discricionárias.

- Modelo de Supereficiência (MSE). Na DEA original, quando várias empresas possuem o escore de eficiência igual a 1, não se pode distinguir hierarquicamente quais são as DMUs mais ou menos eficientes. Deste modo, o MSE faz este ranqueamento.
- Modelo FDH (*Free Disposal Hull*). Foi proposto por Cherchye et. al. (2000) e sugere o relaxamento da restrição da convexidade, quando é empregado um modelo de programação linear inteira mista, com a criação de uma fronteira eficiente de possibilidades, em forma de escada, com variações discretas.
- Modelo de variáveis categóricas ou *Dummies*. É semelhante ao MME, diferindo, apenas por empregar variações temporais nas variáveis.
- Entre outros.

Com o passar do tempo, novos modelos da DEA vão surgindo. Uma fonte atualizada com vários modelos da DEA pode ser encontrada, no sítio DEAZONE.

2.5 Comentários

Este capítulo apresentou todo um arcabouço teórico que suportará o alcance do objetivo deste trabalho. Assim, foi apresentado um histórico da programação linear, junto com alguns de seus fundamentos e o funcionamento do algoritmo Simplex. Em seguida, foi mostrado um histórico da DEA, passando pelos conceitos básicos; seus modelos, com seus formalismos e orientações, além de outros modelos complementares.

Capítulo 3

Este capítulo trata da descrição do levantamento dos dados, utilizados para estimar a fronteira de eficiência com a DEA. Ainda é mostrada uma filtragem nos dados, obtidos pelo método da pesquisa, através de formulários entregues aos ISPs, denominado análise de correlação. Este, ainda apresenta uma descrição detalhada da metodologia empregada, além da arquitetura proposta neste trabalho. A segunda parte do capítulo trata da normalização dos dados colhidos através da matriz de correlação. A terceira parte é remetida à aplicação dos dados, normalizados às formulas da DEA, segundo seus modelos, formas e orientações no software LINDO; além dos cálculos dos alvos (valores a serem atingidos pelas DMUs, nas entradas e saídas, com o intuito de que estas sejam produtivas e eficazes), movimentos radiais, folgas (excessos) das entradas e deficiências (faltas) das saídas.

3.1. Cenário

Os dados foram coletados através de formulários a serem preenchidos pelos responsáveis e/ou proprietários dos ISPs, localizados nas cidades do sul do interior (cariri) do Ceará, mais especificamente nas cidades de Crato e Juazeiro do Norte, além de alguns ISPs, localizados na cidade de Cajazeiras no alto-sertão da Paraíba.

3.1.1. Dados Reais

A avaliação dos ISPs se deu, através de questionários entregues às referidas empresas que as responderam com dados estimativos em cidades do interior do Ceará e Paraíba, cidades estas dispostas na Figura 3.1. O questionário, inicialmente, continha três perguntas, relacionadas às entradas/insumos (Planos Vendidos, Número de Clientes por Plano e a Velocidade Média Registrada por Cliente por Plano); e três perguntas referentes às saídas/produtos (Número de Servidores de Rede, *Link* Contratado e a Estimativa de Lucro Mensal).



Figura 3.1. Mapa dos estados do Ceará e Paraíba com as cidades dos ISPs analisados.

Ao serem aplicados os questionários, observou-se que alguns dos ISPs não ofereciam um número de planos que outros concorrentes ofereciam. Ainda surgiu a possibilidade de se lidar com ISPs que não trabalhavam com tal vazão ou velocidade final para com o cliente. Logo, foi aplicada uma normalização para adequação dos dados das firmas em questão. Cabe, como observação, o fato de que alguns ISPs solicitaram que seus nomes reais não fossem divulgados.

Antes de serem avaliadas as DMUs, as entradas e as saídas passaram por uma matriz de correlação que, segundo Reinaldo et. al.. (2002), serve para medir e avaliar o grau de relação existente entre duas variáveis (x,y), através da disposição destes pontos em torno de uma reta. Este coeficiente de correlação varia de -1 a 1. Assim, a eliminação de variáveis se dá através de uma forte correlação, ou seja, que a relação entre os pontos seja maior que 0,85. Com o emprego da matriz de correlação, nas entradas e saídas originais, foi vislumbrado que algumas variáveis de entrada deveriam ser excluídas, com o intuito de dar mais veracidade na avaliação das DMUs.

Os dados originais, recolhidos pelos questionários estão dispostos na tabela 3.1 com as entradas; e na tabela 3.2, com as saídas. A matriz de correlação dos dados originais será mostrada na tabela 3.3.

Tabela 3.1. Entradas com os dados originais dos ISPs analisados

DMU	Entradas			
	Planos/Usuários			Planos/Média de Vazão
	128	256		128 256
M. Net	30	148		120 205
Baydenet	1000	1000		110 200
ADL Link	150	300		90 154
NetLine	420	1000		120 250
CePlug	60	200		115 200
HotLink	80	180		120 220
Cz.Net	50	150		100 190
PervUbiq	100	250		125 225
Throughput	50	180		120 220

Tabela 3.2. Saídas com os dados originais dos ISPs analisados

DMU	Saídas		
	Servidores	Link Contratado (Mbps)	Lucro Mensal
M. Net	2	2	R\$ 1.400,00
Baydenet	5	50	R\$ 50.000,00
ADL Link	5	30	R\$ 10.000,00
NetLine	2	125	R\$ 40.000,00
CePlug	4	20	R\$ 15.000,00
HotLink	2	30	R\$ 20.000,00
Cz.Net	2	15	R\$ 10.000,00
PervUbiq	3	50	R\$ 35.000,00
Throughput	2	30	R\$ 20.000,00

Tabela 3.3. Matriz de correlação dos dados originais.

	X1	X2	X3	X4	Y1	Y2	Y3
X1	1	-	-	-	-	-	-
X2	0,89	1	-	-	-	-	-
X3	-0,08	0,03	1	-	-	-	-
X4	0,07	0,32	0,87	1	-	-	-
Y1	0,51	0,3	-0,54	-0,63	1	-	-
Y2	0,46	0,79	0,24	0,61	-0,06	1	-
Y3	0,8	0,82	0,3	0,49	0,26	0,73	1

Os dados normalizados serão apresentados ainda neste capítulo. Como pode ser visto, na tabela acima, existem variáveis com o fator de correlação superior a 0,85; ou seja, deverão ser removidas, para dar mais acurácia à avaliação realizada pela DEA.

3.2. Metodologia

Este trabalho utiliza DEA ou Análise Envoltória de Dados, com o objetivo de se estimar uma fronteira Pareto-eficiente das DMUs ou firmas ou empresas analisadas que, no caso, foram os ISPs do cariri cearense e alto-sertão paraibano.

Conforme apresentado no capítulo 2, a DEA é um método econômico não-paramétrico de estimação de fronteira Pareto-Koopmans eficiente, tendo como premissa a avaliação de

unidades de tomada de decisão ou DMUs, ou firmas, entre outros. O pressuposto parte de que, para se produzir um determinado produto, precisa-se da transformação de dada quantidade de insumos em produtos. Como visto na DEA, as unidades não precisam, necessariamente, ser iguais, daí uma grande vantagem da DEA sobre outras metodologias.

A DEA, por sua vez, possui vários estágios para o cálculo da fronteira eficiente, que são:

1. Primeiro estágio – é aplicado o modelo original da DEA e suas respectivas orientações e formas, i.e., é usado o modelo CCR, na forma dos multiplicadores e de envoltórios, orientados a entradas e as saídas, onde as folgas são calculadas residualmente.
2. Segundo estágio – de acordo com [FERREIRA e GOMES 2009] consiste em maximizar a soma das folgas, pois se torna necessário determinar possíveis folgas diferentes de zero, após o primeiro estágio aplicado, ou seja, determinar se uma dada DMU é ou não falso eficiente em qualquer orientação. Um modelo de segundo estágio é o BCC em suas formas e orientações.
3. Múltiplos estágios – de acordo com Coelli (1996), esta abordagem demanda mais desperdício de esforço computacional, mas em compensação identifica os pontos eficientes tanto de entradas, quanto de saídas, i.e., nas DMUs ineficientes e falso eficientes as várias execuções dos problemas de programação linear levariam a determinação dos pontos eficientes, fazendo com que as DMUs em questão tornassem eficientes. Um modelo de múltiplos estágios é o Modelo Aditivo de Folgas ou MBF ou Modelo Baseado em Folgas.

Todos estes estágios foram utilizados por este trabalho com o intuito de, não só projetar saber quais das empresas avaliadas seriam eficientes, mas, também, para se saber identificar algumas DMUs, consideradas falso eficientes, que são aquelas que possuem folgas diferentes de zero e, assim, fazer com que todas as empresas avaliadas possam se tornar eficientes.

Além disto, este trabalho visa a não só aumentar a satisfação do cliente que contrata os serviços prestados pelos ISPs, mas também visa a aumentar a lucratividade dos ISPs, atuando, assim, em ambas as vertentes da DEA, i.e.; tanto na orientação a insumos, como na orientação aos produtos.

3.2.1. Arquitetura proposta

A arquitetura proposta por este trabalho visa, primeiramente, à coleta das informações julgadas primordiais para a solução dos problemas de programação linear, sobre os vários modelos, formas e orientações da DEA, em seus vários estágios.

Tais informações foram coletadas com base em formulários dirigidos, ou seja, as dúvidas dos entrevistados foram minimizadas com a presença do autor deste trabalho, para que as perguntas fossem respondidas a contento, no intuito de uma avaliação melhorada por parte da DEA.

Assim o **levantamento dos dados**, junto às empresas de provimento de serviços de conexão para com a Internet, faz parte da área externa da arquitetura, sendo uma das camadas mais importantes da mesma. Vale ser externado que muitas empresas, das localidades citadas, se negaram a responder o questionário, pois, ou desconhecem o caráter acadêmico, ou esperavam que o autor deste fosse mais um concorrente, atrás de informações estratégicas, para alavancar o seu negócio. A arquitetura pode ser vista na Figura 3.2 (abaixo).

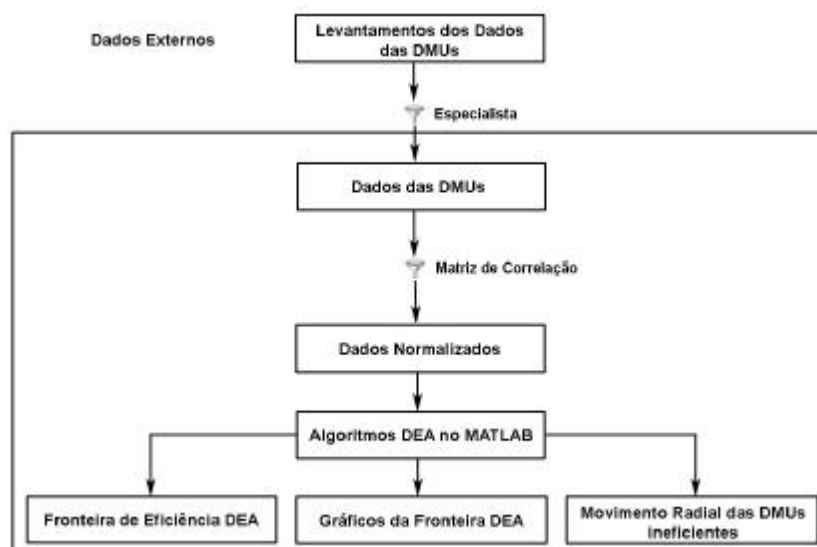


Figura 3.2. Arquitetura proposta pelo trabalho.

Antes de os dados serem inseridos nas camadas internas da arquitetura, foram filtrados por um especialista em redes de computadores para avaliar, primariamente, os dados recebidos dos ISPs, para a geração dos dados que farão parte da segunda camada da arquitetura, que são os dados das companhias a serem avaliadas. Estes dados foram mostrados nas tabelas 3.1 e 3.2; além do mais, vários ISPs não possuíam os planos ou velocidades que outros concorrentes ofereciam, daí outra razão para se efetuar tal normalização dos dados.

O segundo filtro aplicado é a **matriz de correlação** que analisa o grau de relacionamento que uma variável tem, comparado com outra no modelo; quais sejam, entrada com entrada, entrada com saída ou vice-versa. Neste caso, a tabela 3.3 mostrou uma matriz de correlação, onde as variáveis com fator de correlação acima de 0,85, que deveriam ser excluídos para aumentar a precisão da análise obtida com a DEA. Tal matriz de correlação será apresentada em seguida.

A terceira camada são os **dados normalizados**, obtidos através da matriz de correlação. Estes dados foram usados para obtenção da fronteira de eficiência, calculada pela DEA, e serão apresentados adiante.

A quarta camada é a camada da inteligência, i.e., onde são calculadas as fronteiras de eficiência, segundos os modelos, suas formas e orientações, nos primeiro, segundo e múltiplos estágios. Tal camada é referida aos **algoritmos da DEA**, desenvolvidos na linguagem matemática, baseada em matrizes MATLAB, que conta com uma série de funções matemáticas prontas; dentre elas, a linprog que resolve problemas de programação linear, através do algoritmo Simplex e que são obrigatórios, ao se usar a metodologia DEA, para a obtenção da fronteira de eficiência.

As últimas camadas são geradas pela quarta camada, como forma de apresentação ou saída dos dados obtidos, através da quarta camada, pois nestas camadas inferiores estão a **fronteira de eficiência** junto com as DMUs, que são consideradas como *benchmarks*; as ineficientes e as falso eficientes e suas respectivas folgas; os **gráficos de ranqueamento da fronteira DEA** e os **movimentos radiais**, para tornarem as DMUs ineficientes e falso eficientes em eficientes, seguindo seus respectivos parceiros de referência.

Da mesma forma que uma arquitetura foi proposta para embasar o trabalho foi sugerido um fluxograma algorítmico, a fim de mostrar a organização dos dados, usados nesta dissertação, que pode ser visto na Figura 3.3.

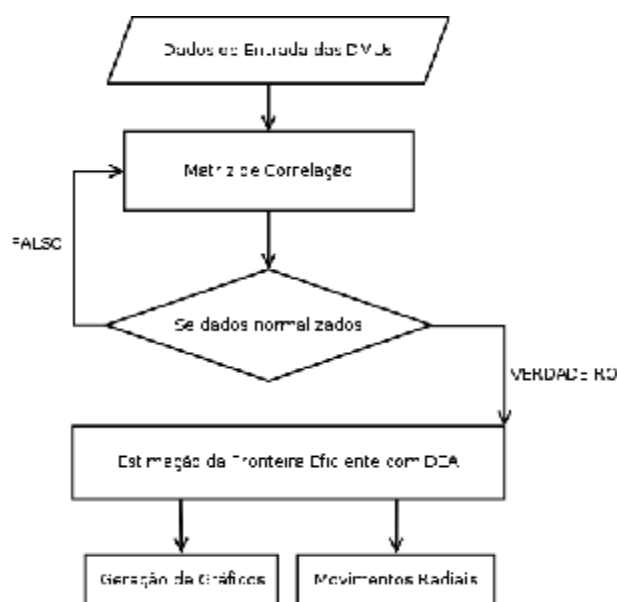


Figura 3.3. Fluxograma algorítmico para estimação de uma fronteira eficiente DEA.

3.3. Normalização dos Dados

Como citado anteriormente, os dados foram normalizados ou filtrados, a partir do cálculo de matrizes de correlação, calculadas a partir das variáveis de entrada e saída. A primeira normalização foi apresentada na tabela 3.3, onde foram excluídas as variáveis que apresentavam fator de correlação acima de 0,85.

A segunda, e definitiva normalização, calculada através do aplicativo de planilha eletrônica OpenCalc, da suíte de aplicativos OpenOffice, poderá ser apreciada na tabela 3.4, a seguir:

Tabela 3.4. Matriz de correlação definitiva dos dados das DMUs.

	X1	X2	Y1	Y2	Y3
X1	1	-	-	-	-
X2	0,35	1	-	-	-
Y1	0,45	-0,45	1	-	-
Y2	0,59	0,65	-0,01	1	-
Y3	0,82	0,49	0,35	0,72	1

Depois destas normalizações ou filtragens realizadas, os dados estão prontos a serem utilizados para calcular a fronteira de eficiência com a DEA, podendo ser visualizados na tabela abaixo.

Tabela 3.5. Dados das DMUs normalizados.

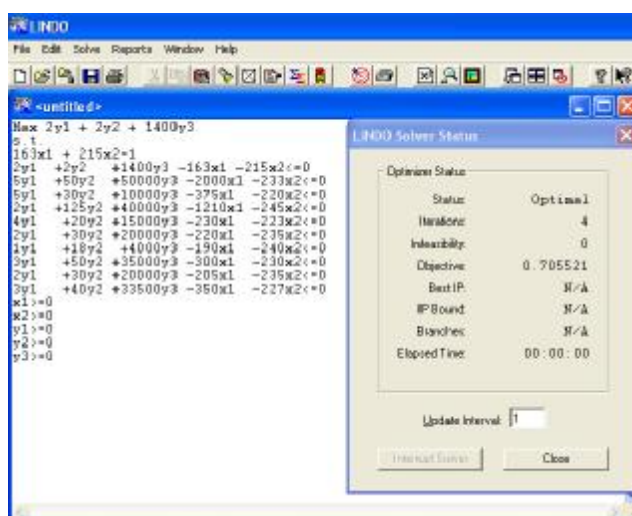
DMU	X1	X2	Y1	Y2	Y3
M. Net	163	215	2	2	R\$ 1.400,00
Baydenet	2000	233	5	50	R\$ 50.000,00
ADL Link	375	220	5	30	R\$ 10.000,00
NetLine	1210	245	2	125	R\$ 40.000,00
CePlug	230	223	4	20	R\$ 15.000,00
HotLink	220	235	2	30	R\$ 20.000,00
Cz.Net	190	240	1	18	R\$ 4.000,00
WiNET	300	230	3	50	R\$ 35.000,00
Throughput	205	235	2	30	R\$ 20.000,00

Onde as entradas são: X1, que é o número de usuários que usam 256 kbps; X2 é a velocidade média registrada pelos clientes. E as saídas são: Y1, que é o número de servidores de rede utilizados pelo provedor de Internet, na localidade pesquisada; Y2 é o *link* contratado pelo ISP e Y3 é a estimativa de lucro mensal.

3.3 Dados normalizados na DEA com seus respectivos cálculos

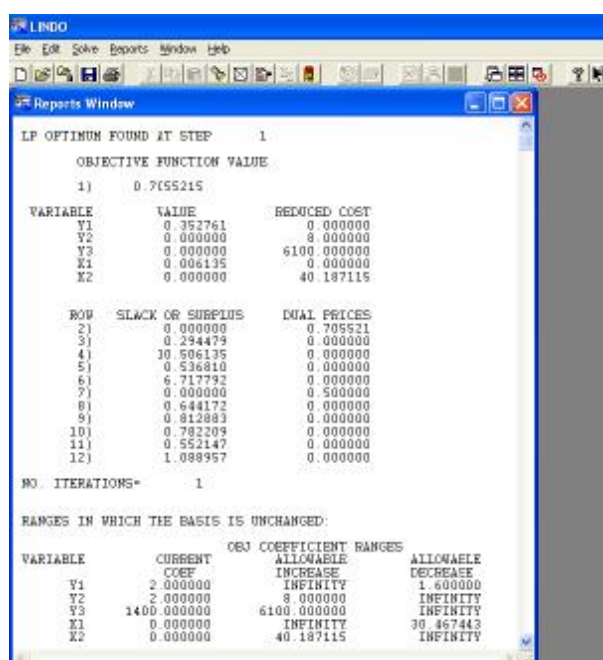
Esta seção foi criada para facilitar o entendimento dos formalismos da DEA, para se estimar as fronteiras de eficiência e, ao mesmo tempo, mostrar a aplicação das fórmulas, apresentadas no capítulo 2, usando o aplicativo LINDO, que calcula valores ótimos para as variáveis através da programação linear, através do uso do método Simplex.

Seguindo a ordem cronológica da DEA será, primeiramente, mostrada uma figura que expõe como se calcular a eficiência de uma DMU. No caso, a M.Net da cidade de Crato-CE, segundo os dados da tabela 3.5 que contém os dados normalizados.

**Figura 3.4. Fórmulas do modelo CCR/M/I aplicados a uma das DMUs analisadas.**

As fórmulas utilizadas foram as (6), (7), (8) e (9) do capítulo 2, que são do modelo CCR, na forma primal ou dos multiplicadores e orientada a insumos ou a entrada. É importante notar que, quando o LINDO soluciona (*solve*) problemas de programação linear, ele exibe uma tela de *status* com um sumário resumido dos cálculos efetuados. Entretanto, o LINDO exibe informações mais detalhadas da resolução dos cálculos, caso seja de interesse do usuário. Assim, tal software exibe estas informações tal como na figura adiante.

É importante notar a rapidez no desempenho computacional, ao se encontrar a solução ótima, através do algoritmo Simplex, ao ser vislumbrado o número de iterações (*iterations*) ou repetições necessárias para se encontrar uma solução ótima no espaço de possibilidades.



LP OPTIMUM FOUND AT STEP 1

OBJECTIVE FUNCTION VALUE

1) 0.7055215

VARIABLE	VALUE	REDUCED COST
Y1	0.352761	0.000000
Y2	0.000000	8.000000
Y3	0.000000	6100.000000
X1	0.006135	0.000000
X2	0.000000	40.187115

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.705521
3)	0.294479	0.000000
4)	10.506135	0.000000
5)	0.536810	0.000000
6)	6.717792	0.000000
7)	0.000000	0.500000
8)	0.644172	0.000000
9)	0.812883	0.000000
10)	0.782209	0.000000
11)	0.552147	0.000000
12)	1.088957	0.000000

NO. ITERATIONS= 1

RANGES IN WHICH THE BASIS IS UNCHANGED:

VARIABLE	CURRENT COEF	OBJ COEFFICIENT RANGES	ALLOWABLE INCREASE	ALLOWABLE DECREASE
Y1	2.000000	INFINITY	INFINITY	1.600000
Y2	2.000000	8.000000	INFINITY	INFINITY
Y3	1400.000000	6100.000000	INFINITY	INFINITY
X1	0.000000	INFINITY	30.467443	INFINITY
X2	0.000000	40.187115	INFINITY	INFINITY

Figura 3.5. Relatório da resolução de problemas de programação linear com o LINDO.

Para se gerar a fronteira de eficiência com a DEA, é preciso repetir tais fórmulas para todas as DMUs a serem analisadas, em todos os modelos, formas e orientações. Contudo, os resultados obtidos no mesmo modelo serão idênticos sob todas as formas e orientações; mas, como exposto anteriormente, as eficiências técnicas do modelo BCC são, geralmente, maiores do que no modelo BCC, onde razão entre a eficiência técnica do modelo CCR pela eficiência técnica do modelo BCC resulta na eficiência de escala (observe a fórmula (20) do capítulo 2).

Em seguida, será mostrado como se calcular a eficiência técnica da DMU, chamada HotLink de Cajazeiras –PB com o LINDO, através do modelo dos multiplicadores, orientação a produtos ou saídas que foram obtidos usando as fórmulas (10), (11) e (12) do capítulo 2.

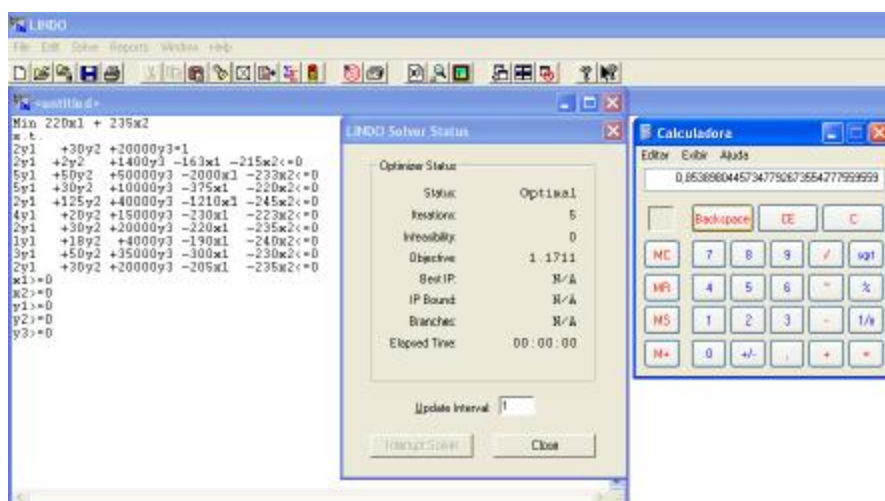


Figura 3.6. Fórmulas do modelo CCR/M/O aplicados a uma das DMUs analisadas.

Relembrando que o resultado da eficiência técnica q será obtido pela divisão entre a unidade (ou 1) por $f = 1.1711$ que, no caso, será igual a 0.8538, como pode ser visto na figura acima.

Abaixo, será ilustrado como se calcular a eficiência técnica, segundo o modelo CCR/E/I. Para fins de prova, a DMU a ser analisada será, novamente, a empresa M. Net, mostrando que os resultados obtidos, através do modelo CCR/M/I; bem como CCR/M/O, serão os mesmos, bastando somente este exemplo como prova, não necessitando da apresentação do modelo CCR/E/O.

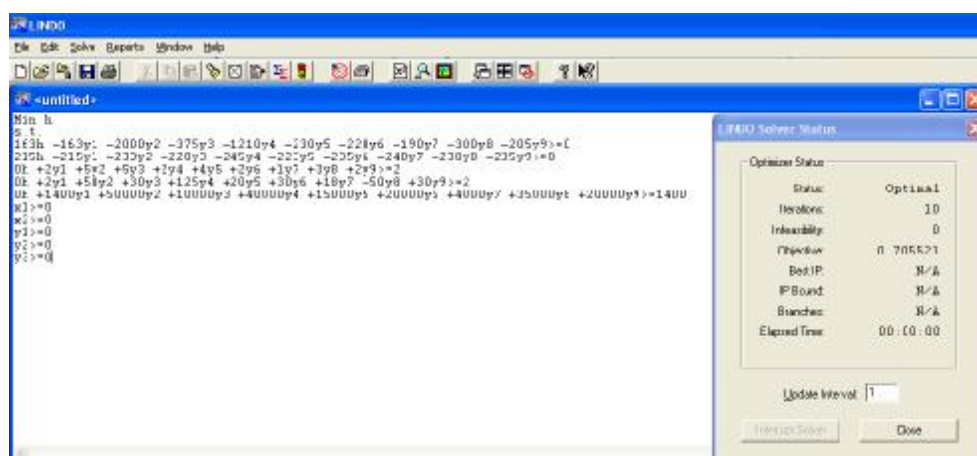


Figura 3.7. Fórmulas do modelo CCR/E/I aplicados a uma das DMUs analisadas.

Os resultados obtidos da DMU M. Net foram obtidos com o uso das fórmulas (13), (14) e (15) do capítulo 2. Uma observação deve ser feita com o uso destas fórmulas, no LINDO, pois pode ser visto que o número de fórmulas é bem menor do que no modelo primal ou de multiplicadores, i.e., no modelo primal o número de inequações ou equações é o número de

DMUs, sendo que, na forma dual ou dos envoltórios (envoltório), é reduzido para o número de variáveis em estudo no modelo.

Para se calcular as folgas e as insuficiências dos produtos em múltiplos estágios, é necessário já possuir a eficiência técnica de uma determinada DMU, a fim de poder multiplicar o valor da eficiência técnica tanto pela entrada, quanto pela saída, obtendo os movimentos radiais em direção à fronteira. Com as soluções dos problemas de programação linear, bem como ao serem visualizadas as fórmulas da DEA, segundo seus modelos, formas e orientações, nota-se a existência de uma variável λ , variável esta que determina os parceiros de referência entre as DMUs, sendo que, com estes valores, são selecionados quais valores de outras DMUs poderão ser usados nos cálculos das folgas e das insuficiências de produtos. De posse desta(s) informação(ões) são calculados os alvos (*targets*) de cada uma das DMUs, assim a diferença entre o movimento radial de cada entrada/saída e os alvos de cada entrada/saída determinará as folgas e as deficiências.

Como exemplo a DMU HotLink de Juazeiro do Norte-CE, com os λ s possuem os seguintes valores: 0,0714 e 0,5714 que correspondem, respectivamente, aos seus parceiros CEPlug e Throughput. Desta forma, os valores destas DMUs deverão ser aplicados para se encontrar os valores tanto dos alvos que, em seguida, serão usados para serem encontradas tanto as folgas, quanto das faltas; além de serem selecionados na matriz que forma as DMUs, somente as linhas e colunas que correspondem aos insumos a serem calculados os alvos. Assim, para se calcular os alvos da DMU supracitada observe as equações abaixo:

Para as entradas:

$$\sum_{i=1}^k l_k x_k, \forall l > 0 \quad (1)$$

$$\sum_{i=1}^k l_k y_k, \forall l > 0 \quad (2)$$

Vale ser observado que o valor de λ tem de ser maior que zero, desta forma a equação (1) calcula os alvos das entradas e a (2), os das saídas. Como exemplo de aplicação das fórmulas acima está o exemplo do cálculo dos alvos de todas as entradas e saídas da DMU HotLink.

$$Alvo_X1 = (0,0714 \times 230) + (0,5714 \times 300)$$

$$Alvo_X2 = (0,0714 \times 223) + (0,5714 \times 230)$$

$$Alvo_Y1 = (0,0714 \times 4) + (0,5714 \times 3)$$

$$Alvo_Y2 = (0,0714 \times 20) + (0,5714 \times 50)$$

$$Alvo_Y3 = (0,0714 \times 20000) + (0,5714 \times 35000)$$

No caso da DMU HotLink de posse do alvo, o cálculo do movimento radial é obtido, através da multiplicação da eficiência técnica (q) da DMU pela entrada/saída em questão. Como a eficiência técnica da DMU HotLink, segundo o CCR é 0,8539, então os valores dos movimentos radiais desta DMU podem ser obtidos através dos cálculos abaixo:

$$Radial_X1 = 0,8539 \times 220$$

$$Radial_X2 = 0,8539 \times 235$$

$$Radial_Y1 = 0,8539 \times 2$$

$$Radial_Y2 = 0,8539 \times 30$$

$$Radial_Y3 = 0,8539 \times 20000$$

Finalmente, para serem calculadas as folgas e as deficiências basta subtrair, para cada entrada e saída, o movimento radial pelo alvo em questão, podendo ser visualizado, em apenas um exemplo a seguir, no caso para se calcular a folga da primeira entrada (X1) e falta (deficiência) da primeira saída (Y1).

$$Folga_X1 = Radial_X1 - Alvo_X1$$

$$Falta_Y1 = Radial_Y1 - Alvo_Y1$$

Enfim, será apresentado o modelo aditivo ou MBF, que, também, como os outros, pode ser resolvido usando o LINDO. Uma peculiaridade, acerca deste modelo, é que ele só possui a forma dual, por conseguinte este modelo também não possui nenhuma orientação, de entrada ou saída. Assim, tal modelo pode ser resolvido através da figura a seguir. Este modelo buscará por falso eficientes, tentando, ao mesmo tempo, diminuir as entradas e maximizar as saídas, buscando os movimentos radiais, visando a eliminação dos excessos dos insumos e as folgas dos produtos.

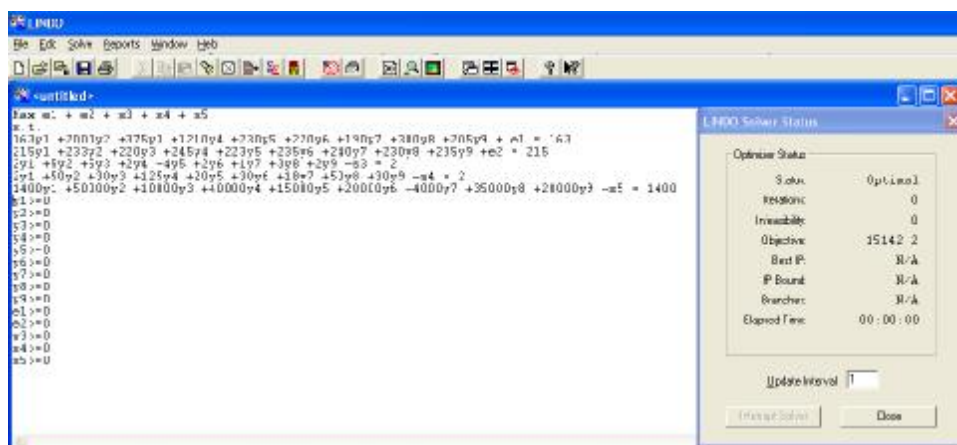


Figura 3.8. Fórmulas do modelo MBF aplicados a uma das DMUs analisadas.

Vale ser observado que o valor da função objetivo é 15142.2, ou seja, o valor da soma das folgas (excessos) e das insuficiências de saídas é o valor acima. Para serem obtidos tais valores, foram usadas as fórmulas (33), (34), (35) e (36) do capítulo 2.

Para esclarecer o valor da função objetivo, faz-se necessário a visualização dos dados de relatórios do LINDO.

The screenshot shows the LINDO Reports Window. It displays the objective function value and the reduced costs for each variable. The table below summarizes the data shown in the report.

Variable	Value	Reduced Cost
E1	0.00000	185.056870
E2	79.87647	0.00000
S1	0.00000	6998.019631
S2	21.17647	0.00000
S3	15041.17675	0.00000
T1	0.00000	15142.239492
T2	0.00000	207381.625000
T3	0.00000	24966.224609
T4	0.00000	-7.250.75625
T5	3.217647	0.00000
T6	0.00000	7119.470707
T7	0.00000	24573.736156
T8	3.376473	0.00000
T9	0.00000	4348.617676

Figura 3.9. Relatório das folgas e faltas do MBF relativo a uma das DMUs analisadas.

É importante notar que, para a DMU M. Net, não existe folga (excesso) da entrada 1, mas existe folga da entrada 2 em 79.876. Em relação às saídas, não existe nenhuma falta no produto 1, mas existe uma falta de 21.176 do produto 2, além de uma falta de 15041.176 para o produto 3. Assim, a soma de todas estas folgas, com estas faltas, resultará no valor de 15142.23.

Outra informação importante é que estes valores são diferentes das folgas e deficiências, calculadas anteriormente, pelo próprio pressuposto básico da MBF apresentado anteriormente.

3.4 Comentários

Este capítulo tratou da descrição do cenário que foi utilizado para estimação de fronteira de eficiência avaliada pela DEA. Desta forma, foram apresentados os dados, na forma com que foram originalmente coletados e que, posteriormente, foram normalizados pela aplicação de uma matriz de correlação. Ainda foi apresentada a metodologia utilizada, no caso DEA, mostrando a arquitetura proposta pelo trabalho, além de um fluxograma algorítmico, utilizado para a confecção básica dos algoritmos que foram desenvolvidos para a realização de todos os cálculos do trabalho. Foram mostrados os dados normalizados pelo uso da técnica de matriz de correlação. Em seguida, estes dados normalizados foram incorporados às diversas fórmulas da DEA, segundo seus modelos, formas e orientações usadas neste trabalho.

Capítulo 4

Resultados e Discussões

Este capítulo apresenta, inicialmente, o ambiente de desenvolvimento onde foi concebido o *Solver* DEA proposto. Em seguida, serão apresentados os resultados, obtidos com a execução do *Solver* DEA, em seus principais modelos: CCR e BCC; formas: multiplicadores e envoltório e orientações: entrada e saída. As seções apresentarão as análises, confrontadas com o Win4DEAP, para mostrar a precisão nos resultados obtidos, além do que algumas análises não implementadas pelo *Solver* DEA proposto são apresentadas, bem como alguns gráficos serão mostrados. Na última seção, são mostrados os resultados obtidos pelo software proposto por este trabalho para análise das DMUs, segundo o MBF.

4.1. Ambiente de Desenvolvimento

O *Solver* DEA proposto foi desenvolvido no MATLAB (MATrix LABoratory) para calcular as fronteiras de eficiência nos principais modelos da DEA, tal como: CCR e BCC, com suas respectivas formas: dos Multiplicadores e de Envoltório, além das orientações de entrada e saída.

Como o MATLAB utiliza as matrizes, como tipo de dados padrão para realizar as operações matemáticas, este trabalho o julgou necessário para se calcular as eficiências das DMUs em análise. O MATLAB ainda possui uma série de funções matemáticas e de outros objetivos já prontas. Dentre estas funções há de se destacar a linprog que usa métodos de programação linear, em busca de valores ótimos, onde pode ser utilizado o método Simplex, usado neste trabalho e na DEA, além do método dos pontos interiores.

Além disso, o MATLAB conta com uma série de ferramentas de geração de gráficos e uma gama de outras funcionalidades, implementadas num pacote simulador, denominado Simulink que, dentre as ferramentas de simulação, podem ser citadas: a simulação mecânica, hidráulica, de potência de sistemas, de eventos e muitas outras. Outro atrativo do MATLAB é uma miríade de ferramentas de cálculo (*Toolboxes*) prontas, tais como: cálculo aeroespacial, lógica fuzzy, comunicações, redes neurais, processamento de imagens, bioinformática, etc.

Outra vantagem do uso do MATLAB é a interoperabilidade com os principais Sistemas Operacionais como Windows, Linux e MAC OS X. Entretanto, um programa desenvolvido neste pode ser traduzido para linguagem C, aumentando ainda mais a portabilidade de um sistema. Outro ponto importante é a capacidade de comunicação com outras linguagens de programação como JAVA e C++, além da orientação a objeto.

A implementação do *Solver* DEA proposto pode ser visualizada nos anexos da dissertação.

4.2. Resultados Obtidos

Nas subseções seguintes serão apresentados os resultados, obtidos pelo software proposto, utilizando DEA em seus principais modelos, formas e orientações.

4.2.1. Resultados das DMUs analisadas segundo o CCR

O CCR é o modelo padrão da DEA, pois foi o primeiro a ser desenvolvido por Charnes, Cooper e Rhodes (1978). Tal modelo é baseado na suposição dos Retornos Constantes de Escala onde as entradas e as saídas são proporcionais entre si. O CCR, originalmente, foi proposto para ser trabalhado, usando a forma fracionária, mas seus autores logo vislumbraram a redução da forma fracionária para um problema de programação linear e, assim, criaram os seus outros dois modelos: o de Multiplicadores (Primal) e de Envoltórios (Dual), ao usar o algoritmo Simplex, para encontrar as fronteiras de eficiência DEA.

Como visto no capítulo anterior, os dados das DMUs analisadas foram obtidos através de questionários, realizados nas empresas pesquisadas, onde nem todas as corporações vendiam os mesmos serviços, assim foram aplicadas as matrizes de correlação duas vezes para aumentar a precisão do julgamento da eficiência das DMUs analisadas. Entretanto, foi aplicada, também, uma normalização nos dados de entradas das DMUs, deixando apenas a velocidade de 256 kbps e o número de clientes deste plano como entradas, permanecendo as saídas inalteradas, pois as matrizes de correlação julgaram desta forma.

Os dados obtidos foram confrontados com o software Win4DEAP, desenvolvido como extensão do DEAP original proposto por Coelli (1996). O Win4DEAP nada mais é do que uma ferramenta visual do DEAP, assim tal software. Além do LINDO foram usados para confrontar a veracidade dos dados do *Solver* DEA proposto, sendo que os dados obtidos, nestas três plataformas, foram os mesmos, atestando, assim, que o *Solver* DEA proposto é realmente funcional para qualquer modelo proposto, em suas formas e orientações.

Como a principal preocupação deste trabalho é aumentar a satisfação dos clientes, este se preocupou, primariamente, em calcular os movimentos radiais, em direção à fronteira de eficiência, as folgas ou excessos das entradas e as deficiências ou faltas das saídas, além dos alvos eficientes no modelo CCR orientado a entradas. Os resultados obtidos serão mostrados abaixo da forma como é a saída original do *Solver* DEA proposto.

===== DEA MATLAB Solver =====																			
===== Eficiência =										= FOLGAS (SLACKS) ==									
===== RADIAL										===== FALTAS ==									
===== ALVOS (TARGETS)										=====									
DMU	Theta	I1	I2	O1	O2	O3	I1	I2	O1	O2	O3	I1	I2	O1	O2	O3	I1	I2	O1
M.Net	0.7055	115.00	151.69	1.41	1.41	987.73	0	40.19	-0	8	6100	115	111.5	2	10	7500			
Baydenet	1.0000	2000.00	233.00	5.00	50	50000	0	0	-0	-0	-0	2000	233	5	50	50000			
ADLLink	1.0000	375.00	220.00	5.00	30	10000	0	0	-0	-0	-0	375	220	5	30	10000			
NetLine	1.0000	1210.00	245.00	2.00	125	40000	0	0	-0	-0	-0	1210	245	2	125	40000			
CEPlug	1.0000	230.00	223.00	4.00	20	15000	0	0	-0	-0	-0	230	223	4	20	15000			
HotLink	0.8539	187.86	200.67	1.71	25.62	17077.92	0	53.31	-0	-0	1071.43	187.86	147.36	2	30	21071.43			
Cz.Net	0.5684	108.00	136.42	0.57	10.23	2273.68	0	53.62	0.08	-0	8600	108	82.80	1.08	18	12600			
WiNet	1.0000	300.00	230.00	3.00	50	35000	0	0	-0	-0	-0	300	230	3	50	35000			
Throughput	0.9164	187.86	215.35	1.83	27.49	18327.53	0	67.99	-0	0	1071.43	187.86	147.36	2	30	21071			

Figura 4.1. Saída do software proposto para o CCR/E/I.

Vale ainda ressaltar que o software proposto está sob constantes melhorias para proporcionar resultados mais inteligíveis. Entretanto, pode-se observar que as DMUs: M.Net, HotLink, Cz.Net e Throughput não são eficientes, visto que necessitam diminuir, proporcionalmente, em 29,45%, 14,61%, 43,16% e 8,36%, respectivamente, suas entradas com o intuito de tornarem-se eficientes, através do movimento radial. Mesmo assim, estas diminuições proporcionais sobre estas DMUs podem acarretar que ainda continuem ineficientes, pois ainda possuem, ora folgas de entrada, ora escassez de saídas, fazendo com que o software proposto tenha de calcular estes valores com a finalidade de alcançar os alvos considerados ótimos.

Ainda pode ser observado que as empresas CEPlug e WiNET são exemplos de excelência, ou seja, um *benchmark* das empresas M.Net, HotLink, CzNet e Throughput. É notório afirmar que, quanto maior o valor positivo de λ (WiNET), mais importante é a DMU eficiente como parceiro de referência; além disso, quanto mais vezes uma DMU eficiente é identificada como parceiro de excelência para DMUs ineficientes, maior é sua importância entre as DMUs eficientes [FERREIRA e GOMES 2009].

DMU	Parceiro(s) de Excelência	Lambdas
1 - M.Net	5.00	0.50
2 - Baydenet	2.00	1.00
3 - ADLLink	3.00	1.00
4 - NetLine	4.00	1.00
5 - CEPlug	5.00	1.00
6 - HotLink	5.00 8.00	0.07 0.57
7 - CzNet	8.00	0.36
8 - WiNet	8.00	1.00
9 - Throughput	5.00 8.00	0.07 0.57

Figura 4.2. Parceiros de Referência (Benchmrks) das DMUs.

Como mostrado anteriormente, os resultados do software proposto foram comparados ao Win4DEAP. Assim, um relatório do Win4DEAP será mostrado abaixo para fins de comparação. Na figura 4.4, será apresentado o sumário de eficiência técnica; na figura 4.5, (a)

têm-se o sumário de folgas das entradas, na figura 4.5 (b) é mostrado o sumário de escassez das saídas; e a figura 4.5 apresenta o sumário dos alvos das entradas (a) e das saídas (b).

EFFICIENCY SUMMARY:

firm	te
1	0.706
2	1.000
3	1.000
4	1.000
5	1.000
6	0.854
7	1.000
8	1.000
9	0.916

Figura 4.3. Sumário de Eficiências Técnicas das DMUs calculadas no Win4DEAP.

SUMMARY OF INPUT SLACKS:

firm	input:	1	2
1		0.000	56.961
2		0.000	0.000
3		0.000	0.000
4		0.000	0.000
5		0.000	0.000
6		0.000	62.430
7		0.000	0.000
8		0.000	0.000
9		0.000	74.196

SUMMARY OF OUTPUT SLACKS:

firm	output:	1	2	3
1		0.000	11.339	8646.087
2		0.000	0.000	0.000
3		0.000	0.000	0.000
4		0.000	0.000	0.000
5		0.000	0.000	0.000
6		0.000	0.000	1254.753
7		0.000	0.000	0.000
8		0.000	0.000	0.000
9		0.000	0.000	1169.282

Figura 4.4. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas.

SUMMARY OF INPUT TARGETS:

firm	input:	1	2
1		163.000	158.039
2		2000.000	233.000
3		375.000	220.000
4		1210.000	245.000
5		230.000	223.000
6		220.000	172.570
7		190.000	240.000
8		300.000	230.000
9		205.000	160.804

SUMMARY OF OUTPUT TARGETS:

firm	output:	1	2	3
1		2.835	14.174	10630.435
2		5.000	50.000	50000.000
3		5.000	30.000	10000.000
4		2.000	125.000	40000.000
5		4.000	20.000	15000.000
6		2.342	35.133	24676.806
7		1.000	18.000	40000.000
8		3.000	50.000	35000.000
9		2.183	32.738	22994.297

Figura 4.5. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas.

Como pôde ser visto, os resultados do Solver DEA proposto, comparados com os resultados do Win4DEAP, são os mesmos para o CCR/E/I.

Os gráficos, gerados pelo *Solver* DEA proposto, apenas mostra as eficiências técnicas das DMUs e pode ser visto na figura abaixo.

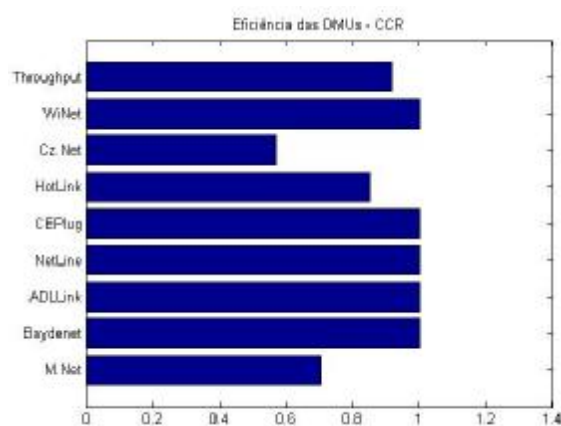


Figura 4.6. Gráfico das Eficiências Técnicas do modelo CCR gerado pelo Solver DEA proposto.

Foi utilizado o DEA-Solver da Saitech, uma versão acadêmica, desenvolvida na planilha eletrônica Excel, utilizando macros para ranquear as DMUs analisadas, além da geração de um gráfico com as eficiências técnicas do modelo CCR que serão mostrados na tabela e figura a seguir.

Tabela 4.1. Ranqueamento das DMUs analisadas segundo CCR/E/I usando o DEA-Solver da Saitech.

Rank	DMU	Score
1	WiNET	1
1	CePlug	1
1	Baydenet	1
1	ADL Link	1
1	NetLine	1
6	Throughpu	0,916376
7	HotLink	0,853896
8	M. Net	0,705521
9	Cz.Net	0,568421

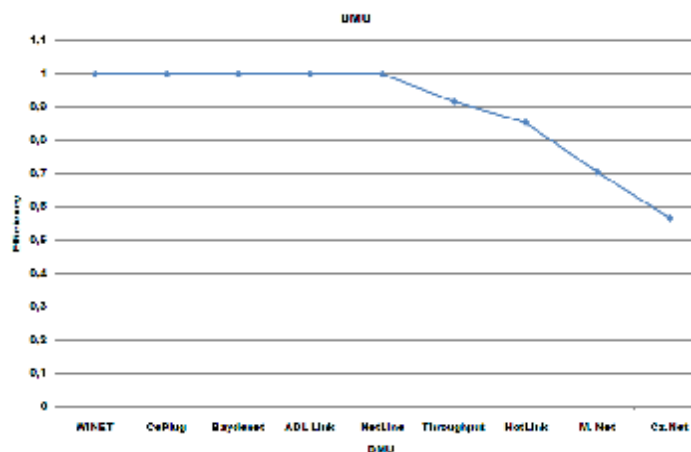


Figura 4.7. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.

As folgas de entradas, faltas das saídas e os alvos, usando o CCR/E/O, foram calculados usando o Win4DEAP, pois o *Solver* DEA proposto só calcula as eficiências técnicas deste modelo. Assim, os resultados são sumarizados nas figuras a seguir, sendo que as eficiências técnicas não serão mostradas, pois são as mesmas mostradas anteriormente. Apesar das eficiências técnicas serem idênticas, as folgas, faltas e alvos são diferentes do modelo CCR/E/I.

SUMMARY OF OUTPUT SLACKS:				SUMMARY OF INPUT SLACKS:				
firm	output:	1	2	3	firm	input:	1	2
1		0.000	11.339	8646.087	1		0.000	56.961
2		0.000	0.000	0.000	2		0.000	0.000
3		0.000	0.000	0.000	3		0.000	0.000
4		0.000	0.000	0.000	4		0.000	0.000
5		0.000	0.000	0.000	5		0.000	0.000
6		0.000	0.000	1254.753	6		0.000	62.430
7		0.000	0.000	0.000	7		0.000	0.000
8		0.000	0.000	0.000	8		0.000	0.000
9		0.000	0.000	1169.202	9		0.000	74.196

Figura 4.8. (a) Sumário das faltas das saídas e (b) Sumário das folgas das entradas, segundo o CCR/E/O.

SUMMARY OF OUTPUT TARGETS:				SUMMARY OF INPUT TARGETS:				
Firm	output:	1	2	3	Firm	input:	1	2
1		2.835	14.174	10630.435	1		163.000	158.039
2		5.000	50.000	50000.000	2		2000.000	233.000
3		5.000	30.000	10000.000	3		375.000	220.000
4		2.000	125.000	40000.000	4		1210.000	245.000
5		4.000	20.000	15000.000	5		230.000	223.000
6		2.342	35.133	24676.806	6		220.000	172.570
7		1.000	18.000	40000.000	7		190.000	240.000
8		3.000	50.000	35000.000	8		300.000	230.000
9		2.183	32.738	22994.297	9		205.000	160.804

Figura 4.9. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o CCR/E/O.

4.2.2 Resultados das DMUs analisadas segundo o BCC

O modelo BCC foi proposto por Banker, Charnes e Cooper, em 1984, com a premissa dos Retornos Variáveis de Escala, que são gerados através de competições imperfeitas, restrições de capital, regulamentações governamentais, entre outros.

Segundo Ferreira e Gomes (2009), as combinações possíveis entre os valores dos insumos e produtos dos ISPs são as mais variadas possíveis, o que dificulta a conclusão sobre a eficiência técnica em que elas operam e quais poderiam ser um modelo operacional que pudesse ser referência para as demais. Desta forma, algumas empresas oferecem planos com velocidades maiores que seus concorrentes; outras possuem um número maior de clientes; outras têm um *link* contratado maior que as outras; outras detêm um maior número de servidores e outras conseguem obter uma maior lucratividade. Todas estas combinações descritas só podem ser

melhores visualizadas com os cálculos dos escores de eficiência técnica dos modelos DEA, além do pressuposto das competições imperfeitas.

O BCC, tal como o CCR, possuem as mesmas formas e orientações, sendo que os resultados das eficiências técnicas, alcançadas através deste, são, geralmente, maiores que os do CCR. Assim, através da fórmula (20) do capítulo 2 é calculada a eficiência de escala que resulta em um nível de eficiência mais adequado de produção.

Os gráficos gerados pelo BCC diferem dos gerados pelo CCR, pois são não-radiais, curvilíneos e convexos, onde no CCR são lineares formando ângulos de 45°.

O *Solver* DEA proposto não calcula as folgas, faltas e alvos no modelo BCC/E/I e nem no BCC/E/O. Desta forma, foi usado o Win4DEAP para sumarizar tais resultados, apresentados nas figuras abaixo.

SUMMARY OF INPUT SLACKS:			SUMMARY OF OUTPUT SLACKS:		
firm	input:		firm	output:	
1	0.000	0.000	1	0.000	0.000
2	0.000	0.000	2	0.000	0.000
3	0.000	0.000	3	0.000	0.000
4	0.000	0.000	4	0.000	0.000
5	0.000	0.000	5	0.000	0.000
6	0.000	0.000	6	0.181	309.607
7	0.000	0.000	7	0.000	0.000
8	0.000	0.000	8	0.000	0.000
9	0.000	0.000	9	0.000	0.000

Figura 4.10. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/I.

SUMMARY OF INPUT TARGETS:			SUMMARY OF OUTPUT TARGETS:		
firm	input:		firm	output:	
1	163.000	215.000	1	2.000	1400.000
2	2000.000	233.000	2	5.000	50000.000
3	375.000	220.000	3	5.000	10000.000
4	1210.000	245.000	4	2.000	40000.000
5	230.000	223.000	5	4.000	15000.000
6	216.739	231.517	6	2.181	20309.607
7	190.000	240.000	7	1.000	40000.000
8	300.000	230.000	8	3.000	35000.000
9	205.000	235.000	9	2.000	20000.000

Figura 4.11. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas, segundo BCC/E/I.

O BCC/E/ também foi calculado o Win4DEAP e, assim os resultados dos cálculos das folgas, faltas e alvos serão mostrados nas figuras a seguir.

SUMMARY OF INPUT SLACKS:			SUMMARY OF OUTPUT SLACKS:		
firm	input:		firm	output:	
1	0.000	0.000	1	0.000	0.000
2	0.000	0.000	2	0.000	0.000
3	0.000	0.000	3	0.000	0.000
4	0.000	0.000	4	0.000	0.000
5	0.000	0.000	5	0.000	0.000
6	0.000	0.994	6	0.000	286.624
7	0.000	0.000	7	0.000	0.000
8	0.000	0.000	8	0.000	0.000
9	0.000	0.000	9	0.000	0.000

Figura 4.12.(a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/O.

SUMMARY OF OUTPUT TARGETS:			SUMMARY OF INPUT TARGETS:		
firm	output:		firm	input:	
1	2.000	2.000	1	163.000	215.000
2	5.000	50.000	2	2000.000	233.000
3	5.000	30.000	3	375.000	220.000
4	2.000	125.000	4	1210.000	245.000
5	4.000	20.000	5	230.000	223.000
6	2.191	32.866	6	220.000	234.006
7	1.000	18.000	7	190.000	240.000
8	3.000	50.000	8	300.000	230.000
9	2.000	30.000	9	205.000	235.000

Figura 4.13. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o BCC/E/O.

O resultado dos cálculos das eficiências técnicas dos modelos CCR e BCC foram confrontados com o intuito de serem calculadas as eficiências de escala das DMUs analisadas, além disso foram calculados os RVE, para se saber se os retornos são constantes, crescentes (ou não decrescente, i.e.; ≥ 0) ou não-crescentes (ou decrescente, i.e.; ≤ 0). Tais resultados são mostrados na tabela adiante.

De acordo com Ferreira e Gomes (2009) os padrões mais altos de eficiência técnica pura (RVE) devem estar combinados com escalas de produção mais efetivas, ou seja, devem ser capazes de gerar maiores margens de lucro. Outro objetivo da eficiência operacional das DMUs é que elas atinjam o que a DEA denomina como *Most Productive Scale Size* – MPSS, i.e., a mais eficiente escala de produção com rendimentos constantes que são custos operacionais constantes. Assim, pode ser observado que as variáveis, utilizadas como entradas e saídas das DMUs que possuem rendimentos constantes, são inalteradas, o que significa serem efetivamente eficientes e eficazes sem possuir folgas ou excessos, para atingir a posição MPSS, o que torna estas DMUs como parceiras de referência ou *benchmarks*.

As variáveis das DMUs com rendimentos decrescentes são reduzidas, assim a quantidade de insumos diminuem em uma proporção maior do que os produtos para atingir a posição MPSS. Por outro lado, quando os rendimentos são crescentes, as DMUs terão de fazer

ajustes diversos nas suas variáveis de entradas e de saídas com o intuito de atingirem a posição MPSS.

Tabela 4.2. Eficiências técnicas, de escala e RVE's das DMUs analisadas.

DMU's	Eficiências			RNC	RND	Tipo Rendimento
	CCR	BCC	Efic. de Escala	≤ 0	≥ 0	
M. Net	0,7055	1,0000	0,7055	0,7055	1,0000	CRESCENTE
Baydenet	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
ADL Link	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
NetLine	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
CePlug	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
HotLink	0,8539	0,9852	0,8667	0,8539	0,9852	DECRESCENTE
Cz.Net	0,5684	0,9842	0,5775	0,5684	1,0000	DECRESCENTE
WiNET	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
Throughput	0,9164	1,0000	0,9164	0,9164	1,0000	CRESCENTE

O *Solver* DEA mostra, graficamente, as eficiências técnicas do modelo BCC em forma de colunas, que pode ser visto adiante.

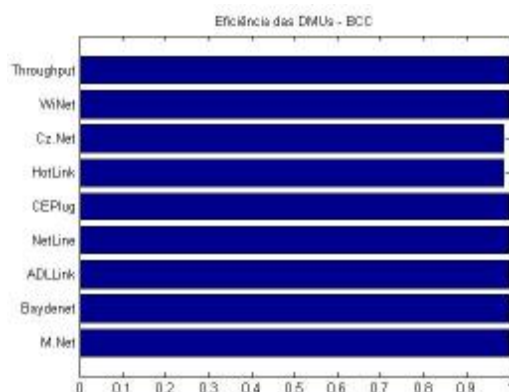


Figura 4.14. Gráfico das Eficiências Técnicas do modelo BCC gerado pelo *Solver* DEA proposto.

No DEA-Solver da Saitech foi gerado um ranqueamento das DMUs, segundo o BCC, apresentado na tabela abaixo.

Tabela 4.3. Ranqueamento das DMUs analisadas segundo BCC/E/I usando o DEA-Solver da Saitech.

Rank	DMU	Score
1	Throughput	1
1	M. Net	1
1	Baydenet	1
1	ADL Link	1
1	NetLine	1
1	CePlug	1
1	WiNET	1
8	HotLink	0,98517
9	Cz.Net	0,984202

O DEA-Solver gerou um gráfico com as eficiências técnicas do BCC que será mostrado em seguida.

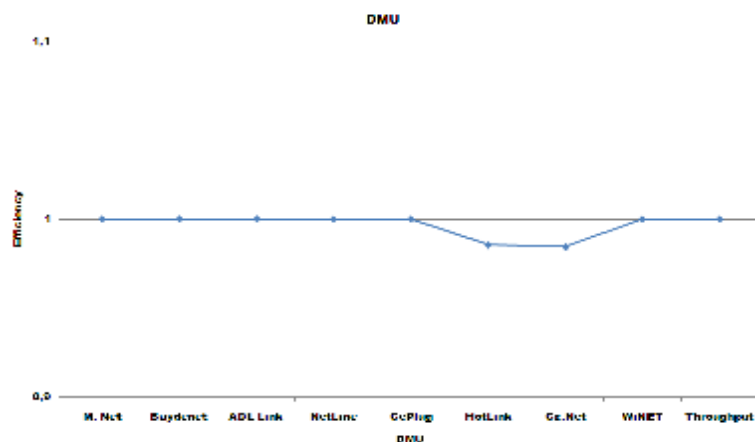


Figura 4.15. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.

4.2.3 Resultados das DMUs analisadas segundo o MBF

O MBF ou Modelo Baseado em Folgas, ou Modelo Aditivo é um modelo especial da DEA que serve para se calcular, simultaneamente, a diminuição nas entradas e o aumento das saídas. Assim, o *Solver* DEA proposto analisou as DMUs, segundo o MBF, e os resultados são mostrados na figura abaixo.

===== DEB MATLAB Solver - MBF =====								
SOMA (FOLGAS + FALTAS)		FOLGAS (INPUT)			FALTAS (OUTPUT)			
DMU		I1	I2	O1	O2	O3		
M.Net	15142.23	0.00	79.88	0.00	21.18	15041.18		
Baydenet	0.00	0.00	0.00	0.00	0.00	0.00		
ADLink	0.00	0.00	0.00	0.00	0.00	0.00		
NetLine	0.00	0.00	0.00	0.00	0.00	0.00		
CePlug	0.00	0.00	0.00	0.00	0.00	0.00		
HotLink	5739.87	0.00	66.33	0.20	6.67	5666.67		
Cz.Net	18275.57	0.00	94.33	0.90	13.67	18166.67		
WiNet	0.00	0.00	0.00	0.00	0.00	0.00		
Throughput	3998.72	0.00	77.83	0.05	4.17	3916.67		

Figura 4.16. Resultados do MBF calculados com o *Solver* DEA proposto.

4.3. Análise dos resultados obtidos

A DEA serviu para comprovar que alguns ISPs oferecem sub-serviços aos usuários finais, em detrimento de seus lucros, quando, principalmente as DMUs: M.Net, HotLink, Cz.Net e Throughput mostraram os piores resultados, comparados com as outras; sendo que as DMUs CePlug e WiNet são consideradas as mais eficientes, pois possuem referências para as DMUs. Assim, estas terão de adaptar seus insumos como os destes benchmarks para se tornarem eficientes.

Por ser um método quantitativo de criação de uma fronteira de eficiência, não-paramétrico, que a DEA precisa de um grande número de DMUs para obter uma eficácia maior nos resultados obtidos. Charnes et. al. (1989) propôs uma heurística que recomenda que o número mínimo de firmas avaliadas deve ser maior ou igual a três vezes a soma das entradas e saídas, outros autores recomendam ainda mais vezes. Assim, este trabalho utilizou um número menor que o mínimo de DMUs, devido ao fato que algumas empresas de provimento de acesso à Internet nas regiões pesquisadas se negaram a responder o questionário, inviabilizando, assim o número mínimo de corporações avaliadas.

A ineficiência ou falsa eficiência deve-se ao fato de que estas têm mais usuários do que, realmente, poderiam ter para oferecer serviços com a velocidade contratada, ou seja, a quantidade de insumos é superior a quantidade necessária para que estas DMUs tornem-se eficientes. Assim, estes insumos devem ser diminuídos, mantendo a mesma proporção de saídas. Outros ISPs tornam-se cada vez mais produtivos, pois estão conectados a seus *backbones*, através de links de fibra ótica, aumentando a qualidade dos serviços oferecidos, onde as velocidades, acordadas em contrato, são quase sempre seguidas ao longo do dia.

4.4. Comentários

Este capítulo mostrou em que plataforma foi desenvolvido o *Solver* DEA proposto, e os motivos levados ao desenvolvimento no MATLAB. Foram apresentados os resultados, obtidos com o uso da DEA, nos principais modelos: CCR e BCC, com suas respectivas formas: dos Multiplicadores e de Envoltório e nas orientações de entrada e saída usando o *Solver* DEA proposto. Além de outros resultados, obtidos com o software Win4DEAP e o DEA-Solver da Saitech, tais softwares foram usados para serem calculados alvos, folgas das entradas, faltas das saídas, ranqueamento e gráficos do CCR/E/O, BCC/E/I e BCC/E/O.

Finalmente, os resultados do MBF das DMUs analisadas foram calculados com o *Solver* DEA proposto e apresentados.

Capítulo 5

Neste capítulo são feitas as considerações finais do trabalho, além da apresentação de melhorias e trabalhos futuros, acerca da pesquisa e/ou relacionada com a DEA.

5.1 Conclusões

Neste trabalho de dissertação foi utilizada uma técnica, baseada no teorema de Pareto-Koopmans e nos métodos de avaliação de produtividade de empresas, desenvolvido por Farrell. Tal técnica foi criada com o intuito de obtenção do grau de PhD de Edwardo Rhodes, em 1978, juntamente com Charnes e Cooper e foi intitulada DEA (Data Envelopment Analysis) ou Análise Envoltória de Dados.

A DEA é uma técnica de geração de fronteira de eficiência Pareto-Koopmans, i.e., onde ora se diminui a quantidade de insumos, para se produzir o mesmo produto; ou ora se aumenta os produtos, mantendo a mesma quantidade de insumos. Além do mais, a DEA consegue estimar sua fronteira de eficiência, com múltiplas entradas e múltiplas saídas, através do algoritmo Simplex de programação linear.

Este trabalho utilizou-se da DEA, para estimar várias fronteiras de eficiências, segundo os seus principais modelos: CCR e BCC, suas formas: dos multiplicadores (primal) e de envoltório (dual) e nas orientações à entrada e à saída; ou seja, o trabalho visa a tanto trabalhar com a satisfação dos clientes, com a melhoria nos serviços por eles contratados, quanto à lucratividade das empresas analisadas.

Vale ser ressaltado que os resultados das eficiências técnicas são idênticos, quando se usa o modelo CCR em qualquer forma ou orientação, bem como no BCC, mas mesmo DMUs, consideradas eficientes, podem apresentar excesso de insumos, denominadas folgas, ou falta de produtos. Logo, deve-se calcular, não só as eficiências técnicas, mas, também, os movimentos radiais para a fronteira de eficiência, as folgas das entradas, as faltas das saídas e os alvos de eficiência na fronteira. Por esta razão é dito que o cálculo é realizado em múltiplos estágios.

O *Solver* DEA proposto foi desenvolvido no MATLAB, uma plataforma de desenvolvimento matemático, baseado em matrizes, onde foi utilizada função linprog, que utiliza o algoritmo Simplex, para resolver problemas de programação linear. Destarte, este implementou todos os principais modelos, formas e orientações da DEA, para estimação da fronteira de eficiência; mas somente calculou os múltiplos estágios no CCR/E/I, pois o principal objetivo deste trabalho era aumentar o grau de satisfação do usuário do serviço. Entretanto, os múltiplos estágios, segundo o CCR/E/O, BCC/E/I e BCC/E/O, foram calculados usando o

software Win4DEAP, sendo que os gráficos de eficiência foram gerados pelo *Solver* DEA proposto, enquanto os gráficos de ranqueamento foram gerados com o DEA-Solver da SAITECH.

O *Solver* DEA proposto ainda calcula em múltiplos estágios, segundo o MBF (Modelo Baseado em Folgas) ou Modelo Aditivo, que pressupõe poder, simultaneamente, diminuir a quantidade de entradas e aumentar a quantidade de produtos produzidos.

Ao rodar todos os modelos, descobriu-se que alguns ISPs ainda trabalham, oferecendo sub-serviços aos clientes, tendo que diminuir, consideravelmente, a quantidade de usuários com o objetivo de atingir os anseios dos consumidores em relação ao contrato firmado entre as partes.

Também foi observado que o preço dos *links* vem caindo, substancialmente, no interior do Nordeste do país, pois as Teles começam a oferecer serviços de interconexão para provedores de serviços da última milha com fibra óptica. Tal benefício, junto com ferramentas de software que dividem a banda, tal como o classificador de pacotes CBQ, no Linux, fazem com que estes provedores possam chegar próximos das velocidades acordadas com os seus *backbones*, porém contratam vários serviços ADSL, tal como o Velox, de 1Mbps para proverem seus serviços de forma mais barata.

No entanto, observou-se a acurácia da técnica DEA que será utilizada pelo autor deste trabalho, para outros fins acadêmicos e que serão apresentados na seção seguinte.

5.2 Trabalhos futuros

Como extensão deste e outros trabalhos futuros são propostos:

- (i) Implementação dos cálculos, em múltiplos estágios da DEA, no CCR/E/O, BCC/E/I e BCC/E/O;
- (ii) Parametrização do *Solver* DEA proposto para que usuários possam usá-lo mais facilmente;
- (iii) Instigar o desempenho do *Solver* DEA proposto com um maior número de DMUs para avaliar seu desempenho em relação a outros softwares DEA;
- (iv) Criação de gráficos de fronteira eficiente no CCR e BCC;
- (v) Implementação de um Web DEA livre e gratuito, para calcular os principais modelos da DEA em múltiplos estágios, com os respectivos gráficos, provendo uma maneira ubíqua de

se estimar fronteiras de eficiência entre empresas, setores de empresas, órgãos governamentais, entre outros;

(vi) Utilização do Modelo de Supereficiência da DEA, no escalonamento de processos de Sistemas Operacionais e/ou em grades computacionais, além de testar a DEA em algoritmos de roteamento em redes *ad-hoc* sem fio.

Referências Bibliográficas

- BARFORD, P.; SOMMERS, J. (2004). Comparing probe-based and router-based packet-loss measurement. *IEEE Internet Computing*, 8(5):50–56.
- CHARNES, A.; COOPER, W.W. & RHODES, E. (1978). Measuring efficiency of decision making units, *European Journal of Operational Research*, 3, pp. 429-444.
- CHARNES, A.; COOPER W.W.; GOLANY, B.; SEIFORD, L.M.; AND STUTZ, J. (1985). “Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions.” *Journal of Econometrics* 30, 91-107.
- CHARNES, A.; COOPER, W. W.; AHN, TAE SIK; ARNOLD, V. (1989). "DEA and Ratio Efficiency Analyses for Public Institutions of Higher Learning in Texas". *Research in Governmental and Nonprofit Accounting* (1989), 5:165-185.
- CHAVES, A. C. A; THOMAZ, A. C. F. (2008). “Gestão Pública e Pesquisa Operacional: avaliação de desempenho em Agências da Previdência Social”, *Revista do Serviço Público*, Brasília 59 (2): 221-236 Abr/Jun 2008.
- CHERCHYE, L.; T. KUOSMANEN and T. POST. (2000). "What is the Economic Meaning of FDH?", *Journal of Productivity Analysis* 13 (3), 263-267.
- COELLI, T. (1996). A guide to DEAP version 2.1: a data envelopment analysis (computer) program. Center for Efficiency and Productivity Analysis. Department of Econometrics, New England University.
- COOPER, W.W.; SEIFORD, L.M. & ZHU, J. (2004). *Data Envelopment Analysis: Models and interpretations*. In *Handbook on Data Envelopment Analysis*, Boston: Kluwer Academic Publisher.
- COOPER, W.W.; SEIFORD, L.M. & TONE, K. (2007). *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Kluwer Academic Publishers, USA.
- DANTZIG, G. (1963). *Linear Programming and Extensions*, Princeton University Press, Princeton.
- ENZO, B. MARIANO; ALMEIDA, R. MARIANA; REBELATO, A. N. DAISY. (2006). “Peculiaridades da Análise Envoltória de Dados”, XII SIMPEP, 2006.

- FERREIRA, CARLOS M. DE CARVALHO; GOMES, PROVEZANO. (2009). "Introdução à Análise Envoltória de Dados". Editora UFV, 1ª edição.
- KARMAKAR, N. (1984). "A New Polynomial Time Algorithm for Linear Programming", *Combinatoria* 4, pp.373-395.
- KOOPMANS, TJALLING CHARLES. (1951). *Activity analysis of production and allocation : proceedings of a conference*, John Wiley and Sons, New York.
- LEVERTY, J.T.; GRACE, M.F. (2007). *Issues in Measuring the Efficiency of Property-Liability Insurers*, Palgrave Macmillan.
- LINS, M. P. E. ; CALÔBA, G.M. (2006). *Programação Linear com aplicações em teoria dos jogos e avaliação de desempenho (Data Envelopment Analysis)*. 1. ed. Rio de Janeiro: Interciência, v. 1. 299 p.
- MACEDO, M. A. S.; SANTOS, Ruthberg dos ; ANDRADE, Jarbas Emanuel Antunes; BENAC, Marcos Azevedo. (2005). *Avaliação da Eficiência dos Investimentos em TI de Empresas de Tecnologia através da Análise Envoltória de Dados (DEA)*. In: VIII SEMEAD - Seminários em Administração da USP, São Paulo, SP. Anais do VIII SEMEAD-USP, 2005.
- NEUMANN, J VON. (1945). A model of general economic equilibrium. *Review of Economic Studies*, XIII,1–9.
- QUINTANILHA, J. A.; HO, L. L. (2006). A performance index developed by data envelopment analysis (DEA) to compare the efficiency of fire risk monitoring actions in municipalities of Brazilian Amazon region. In: *Accuracy 2006: The Seventh International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, 2006, Lisboa, Portugal. The Seventh International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences. Lisboa - Portugal : Instituto Geográfico Português, p. 624-632.
- REINALDO, R. R. P.; POSSAMAI, O.; THOMAZ, A. C. F. (2002). Avaliando a eficiência em unidades de ensino fundamental de Fortaleza usando técnicas de análise envoltória de dados (DEA). *Revista Científica da Faculdade Lourenço Filho*, [S.l.], v. 2, n. 1. Disponível em: <<http://www.flf.edu.br/midias/FLF.EDU/21.pdf>>. Acesso em 30 mar. 2010.
- RESENDE NETO, ANTONIO DE LARA. (2006). *Uma Proposta de Construção de Indicador de Performance de Fundos de Investimento*, UnB, 86 p.
- RAJ JAIN. (1991). "The Art of Computer Systems. Performance Analysis", Wiley & Sons.

VILLARROEL DÁVALOS, R. D.; MORALES, A. T. ; MAGAJEWSKI, F.; SCHLICKMANN, T. (1999) . Uma proposta para modelar os indicadores da saúde através de uma análise envoltória de dados. In: XXXI SBPO - Simpósio Brasileiro de Pesquisa Operacional, Juíz de Fora. XXXI SBPO.

ZIVIANI, ARTUR; DUARTE, O. C. M. B. (2005). Mini-curso do XXIII Simpósio Brasileiro de Redes de Computadores SBRC'2005. Sociedade Brasileira de Computação, 2005.

ANEXOS

% DEA - CCR - Envelopment Mode - Input Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;  
clear;
```

```
f= [1;0;0;0;0;0;0;0;0;]; %objective function
```

```
b_A = [ 0; 0; 2;2;1400]; %constraints
```

```
DMU_A = [ 163 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
215 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_B = [ 0;0; 5;50;50000]; %constraints
```

```
DMU_B = [ 2000 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
233 -215 -233 -220 -245 -223 -235 -240 -230 -235 ; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_C = [ 0;0; 5;30;10000]; %constraints
```

```
DMU_C = [ 375 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
220 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_D = [ 0;0; 2;125;40000]; %constraints
```

```
DMU_D = [ 1210 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
245 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_E = [ 0;0; 4;20;15000]; %constraints
```

```
DMU_E = [ 230 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
223 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_F = [0;0; 2;30;20000]; %constraints
```

```
DMU_F = [ 220 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1  
235 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2  
0 2 5 5 2 4 2 1 3 2; % Output_1  
0 2 50 30 125 20 30 18 50 30; % Output_2  
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_G = [ 0;0; 1;18;4000]; %constraints
```

```
DMU_G = [ 190 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
```

```

240 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

b_H = [ 0;0;3;50;35000;]; %constraints

DMU_H = [ 300 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
230 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

b_I = [ 0;0; 2;30;20000]; %constraints

DMU_I = [205 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
235 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

lb=zeros(10,1);

% Calculation of Technical Efficiency

[lambda_A,FVAL_A,exitflag,output,lambda]=linprog(f,-DMU_A,-b_A,[],[],lb);

[lambda_B,FVAL_B,exitflag,output,lambda]=linprog(f,-DMU_B,-b_B,[],[],lb);

[lambda_C,FVAL_C,exitflag,output,lambda]=linprog(f,-DMU_C,-b_C,[],[],lb);

[lambda_D,FVAL_D,exitflag,output,lambda]=linprog(f,-DMU_D,-b_D,[],[],lb);

[lambda_E,FVAL_E,exitflag,output,lambda]=linprog(f,-DMU_E,-b_E,[],[],lb);

[lambda_F,FVAL_F,exitflag,output,lambda]=linprog(f,-DMU_F,-b_F,[],[],lb);

[lambda_G,FVAL_G,exitflag,output,lambda]=linprog(f,-DMU_G,-b_G,[],[],lb);

[lambda_H,FVAL_H,exitflag,output,lambda]=linprog(f,-DMU_H,-b_H,[],[],lb);

[lambda_I,FVAL_I,exitflag,output,lambda]=linprog(f,-DMU_I,-b_I,[],[],lb);

for i=2:10
    if lambda_A(i,1)>0.01
        disp('O valor DMU_A para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_A(i,1));
        lambdaA(1,i)=lambda_A(i,1);
    end
end

LambdaA_index=find(lambdaA); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaA_index)
    LambdaA_index(1,j)=LambdaA_index(1,j)-1;
end

```



```

end
lambdaA=lambdaA(lambdaA~=0);

for i=2:10
    if lambda_B(i,1)>0.01
        disp('O valor DMU_B para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_B(i,1));
        lambdaB(1,i)=lambda_B(i,1);

    end
end

LambdaB_index=find(lambdaB); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaB_index)
    LambdaB_index(1,j)=LambdaB_index(1,j)-1;
end
lambdaB=lambdaB(lambdaB~=0);

for i=2:10
    if lambda_C(i,1)>0.01
        disp('O valor DMU_C para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_C(i,1));
        lambdaC(1,i)=lambda_C(i,1);

    end
end

LambdaC_index=find(lambdaC); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaC_index)
    LambdaC_index(1,j)=LambdaC_index(1,j)-1;
end
lambdaC=lambdaC(lambdaC~=0);

for i=2:10
    if lambda_D(i,1)>0.01
        disp('O valor DMU_D para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_D(i,1));
        lambdaD(1,i)=lambda_D(i,1);

    end
end

LambdaD_index=find(lambdaD); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaD_index)
    LambdaD_index(1,j)=LambdaD_index(1,j)-1;
end
lambdaD=lambdaD(lambdaD~=0);

for i=2:10
    if lambda_E(i,1)>0.01

```

```

disp('O valor DMU_E para o Lambda ');
disp(i-1);
disp('é:');
disp(lambda_E(i,1));
lambdaE(1,i)=lambda_E(i,1);

end
end

LambdaE_index=find(lambdaE); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaE_index)
    LambdaE_index(1,j)=LambdaE_index(1,j)-1;
end
lambdaE=lambdaE(lambdaE~=0);

for i=2:10
    if lambda_F(i,1)>0.01
        disp('O valor DMU_F para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_F(i,1));
        lambdaF(1,i)=lambda_F(i,1);

    end
end

LambdaF_index=find(lambdaF); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaF_index)
    LambdaF_index(1,j)=LambdaF_index(1,j)-1;
end
lambdaF=lambdaF(lambdaF~=0);

for i=2:10
    if lambda_G(i,1)>0.01
        disp('O valor DMU_G para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_G(i,1));
        lambdaG(1,i)=lambda_G(i,1);

    end
end

LambdaG_index=find(lambdaG); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaG_index)
    LambdaG_index(1,j)=LambdaG_index(1,j)-1;
end
lambdaG=lambdaG(lambdaG~=0);

for i=2:10
    if lambda_H(i,1)>0.01
        disp('O valor DMU_H para o Lambda ');
        disp(i-1);
    end
end

```

```

        disp('é:');
        disp(lambda_H(i,1));
        lambdaH(1,i)=lambda_H(i,1);

    end
end

LambdaH_index=find(lambdaH); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaH_index)
    LambdaH_index(1,j)=LambdaH_index(1,j)-1;
end
lambdaH=lambdaH(lambdaH~=0);

for i=2:10
    if lambda_I(i,1)>0.01
        disp('O valor DMU_I para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_I(i,1));
        lambdaI(1,i)=lambda_I(i,1);

    end
end

LambdaI_index=find(lambdaI); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaI_index)
    LambdaI_index(1,j)=LambdaI_index(1,j)-1;
end
lambdaI=lambdaI(lambdaI~=0);

% Calculation of Slacks

SomaA1=0;
SomaA2=0;
SomaA3=0;
SomaA4=0;
SomaA5=0;
for i=1:length(LambdaA_index)
    SomaA1 = SomaA1 + (lambdaA(1,i)) * (-DMU_A(1,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA2 = SomaA2 + (lambdaA(1,i)) * (-DMU_A(2,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA3 = SomaA3 + (lambdaA(1,i)) * (-DMU_A(3,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA4 = SomaA4 + (lambdaA(1,i)) * (-DMU_A(4,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA5 = SomaA5 + (lambdaA(1,i)) * (-DMU_A(5,LambdaA_index(1,i)+1));
end

inputSlack1_A = (FVAL_A * DMU_A(1,1)) - (SomaA1);

```

```

inputSlack2_A = (FVAL_A * DMU_A(2,1)) - (SomaA2);
outputSlack1_A = (FVAL_A * DMU_A(3,1)) - (SomaA3);
outputSlack2_A = (FVAL_A * DMU_A(4,1)) - (SomaA4);
outputSlack3_A = (FVAL_A * DMU_A(5,1)) - (SomaA5);

```

```

SomaB1=0;
SomaB2=0;
SomaB3=0;
SomaB4=0;
SomaB5=0;

```

```

for i=1:length(LambdaB_index)
    SomaB1 = SomaB1 + (lambdaB(1,i)) * (-DMU_B(1,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB2 = SomaB2 + (lambdaB(1,i)) * (-DMU_B(2,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB3 = SomaB3 + (lambdaB(1,i)) * (-DMU_B(3,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB4 = SomaB4 + (lambdaB(1,i)) * (-DMU_B(4,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB5 = SomaB5 + (lambdaB(1,i)) * (-DMU_B(5,LambdaB_index(1,i)+1));
end

```

```

inputSlack1_B = (FVAL_B * DMU_B(1,1)) - (SomaB1);
inputSlack2_B = (FVAL_B * DMU_B(2,1)) - (SomaB2);
outputSlack1_B = (FVAL_B * DMU_B(3,1)) - (SomaB3);
outputSlack2_B = (FVAL_B * DMU_B(4,1)) - (SomaB4);
outputSlack3_B = (FVAL_B * DMU_B(5,1)) - (SomaB5);

```

```

SomaC1=0;
SomaC2=0;
SomaC3=0;
SomaC4=0;
SomaC5=0;

```

```

for i=1:length(LambdaC_index)
    SomaC1 = SomaC1 + (lambdaC(1,i)) * (-DMU_C(1,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC2 = SomaC2 + (lambdaC(1,i)) * (-DMU_C(2,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC3 = SomaC3 + (lambdaC(1,i)) * (-DMU_C(3,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC4 = SomaC4 + (lambdaC(1,i)) * (-DMU_C(4,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaC5 = SomaC5 + (lambdaC(1,i)) * (-DMU_C(5,LambdaC_index(1,i)+1));
end

```

```

inputSlack1_C = (FVAL_C * DMU_C(1,1)) - (SomaC1);
inputSlack2_C = (FVAL_C * DMU_C(2,1)) - (SomaC2);
outputSlack1_C = (FVAL_C * DMU_C(3,1)) - (SomaC3);

```

```
outputSlack2_C = (FVAL_C * DMU_C(4,1)) - (SomaC4);
outputSlack3_C = (FVAL_C * DMU_C(5,1)) - (SomaC5);
```

```
SomaD1=0;
SomaD2=0;
SomaD3=0;
SomaD4=0;
SomaD5=0;
```

```
for i=1:length(LambdaD_index)
    SomaD1 = SomaD1 + (lambdaD(1,i)) * (-DMU_D(1,LambdaD_index(1,i)+1));
    single(SomaD1);
end
for i=1:length(LambdaD_index)
    SomaD2 = SomaD2 + (lambdaD(1,i)) * (-DMU_D(2,LambdaD_index(1,i)+1));
    single(SomaD2);
end
for i=1:length(LambdaD_index)
    SomaD3 = SomaD3 + (lambdaD(1,i)) * (-DMU_D(3,LambdaD_index(1,i)+1));
end
for i=1:length(LambdaD_index)
    SomaD4 = SomaD4 + (lambdaD(1,i)) * (-DMU_D(4,LambdaD_index(1,i)+1));
end
for i=1:length(LambdaD_index)
    SomaD5 = SomaD5 + (lambdaD(1,i)) * (-DMU_D(5,LambdaD_index(1,i)+1));
end
```

```
inputSlack1_D = (FVAL_D * DMU_D(1,1)) - (SomaD1);
inputSlack2_D = (FVAL_D * DMU_D(2,1)) - (SomaD2);
outputSlack1_D = (FVAL_D * DMU_D(3,1)) - (SomaD3);
outputSlack2_D = (FVAL_D * DMU_D(4,1)) - (SomaD4);
outputSlack3_D = (FVAL_D * DMU_D(5,1)) - (SomaD5);
```

```
SomaE1=0;
SomaE2=0;
SomaE3=0;
SomaE4=0;
SomaE5=0;
```

```
for i=1:length(LambdaE_index)
    SomaE1 = SomaE1 + (lambdaE(1,i)) * (-DMU_E(1,LambdaE_index(1,i)+1));
    single(SomaE1);
end
for i=1:length(LambdaE_index)
    SomaE2 = SomaE2 + (lambdaE(1,i)) * (-DMU_E(2,LambdaE_index(1,i)+1));
    single(SomaE2);
end
for i=1:length(LambdaE_index)
    SomaE3 = SomaE3 + (lambdaE(1,i)) * (-DMU_E(3,LambdaE_index(1,i)+1));
end
for i=1:length(LambdaE_index)
    SomaE4 = SomaE4 + (lambdaE(1,i)) * (-DMU_E(4,LambdaE_index(1,i)+1));
end
for i=1:length(LambdaE_index)
    SomaE5 = SomaE5 + (lambdaE(1,i)) * (-DMU_E(5,LambdaE_index(1,i)+1));
end
```

```
inputSlack1_E = (FVAL_E * DMU_E(1,1)) - (SomaE1);
inputSlack2_E = (FVAL_E * DMU_E(2,1)) - (SomaE2);
```

```

outputSlack1_E = (FVAL_E * DMU_E(3,1)) - (SomaE3);
outputSlack2_E = (FVAL_E * DMU_E(4,1)) - (SomaE4);
outputSlack3_E = (FVAL_E * DMU_E(5,1)) - (SomaE5);

```

```

SomaF1=0;
SomaF2=0;
SomaF3=0;
SomaF4=0;
SomaF5=0;

```

```

for i=1:length(LambdaF_index)
    SomaF1 = SomaF1 + (lambdaF(1,i)) * (-DMU_F(1,LambdaF_index(1,i)+1));
    single(SomaF1);
end
for i=1:length(LambdaF_index)
    SomaF2 = SomaF2 + (lambdaF(1,i)) * (-DMU_F(2,LambdaF_index(1,i)+1));
    single(SomaF2);
end
for i=1:length(LambdaF_index)
    SomaF3 = SomaF3 + (lambdaF(1,i)) * (-DMU_F(3,LambdaF_index(1,i)+1));
end
for i=1:length(LambdaF_index)
    SomaF4 = SomaF4 + (lambdaF(1,i)) * (-DMU_F(4,LambdaF_index(1,i)+1));
end
for i=1:length(LambdaF_index)
    SomaF5 = SomaF5 + (lambdaF(1,i)) * (-DMU_F(5,LambdaF_index(1,i)+1));
end

```

```

inputSlack1_F = (FVAL_F * DMU_F(1,1))- (SomaF1);
inputSlack2_F = (FVAL_F * DMU_F(2,1)) - (SomaF2);
outputSlack1_F = (FVAL_F * DMU_F(3,1)) - (SomaF3);
outputSlack2_F = (FVAL_F * DMU_F(4,1)) - (SomaF4);
outputSlack3_F = (FVAL_F * DMU_F(5,1)) - (SomaF5);

```

```

SomaG1=0;
SomaG2=0;
SomaG3=0;
SomaG4=0;
SomaG5=0;

```

```

for i=1:length(LambdaG_index)
    SomaG1 = SomaG1 + (lambdaG(1,i)) * (-DMU_G(1,LambdaG_index(1,i)+1));
    single(SomaF1);
end
for i=1:length(LambdaG_index)
    SomaG2 = SomaG2 + (lambdaG(1,i)) * (-DMU_G(2,LambdaG_index(1,i)+1));
    single(SomaF2);
end
for i=1:length(LambdaG_index)
    SomaG3 = SomaG3 + (lambdaG(1,i)) * (-DMU_G(3,LambdaG_index(1,i)+1));
end
for i=1:length(LambdaG_index)
    SomaG4 = SomaG4 + (lambdaG(1,i)) * (-DMU_G(4,LambdaG_index(1,i)+1));
end
for i=1:length(LambdaG_index)
    SomaG5 = SomaG5 + (lambdaG(1,i)) * (-DMU_G(5,LambdaG_index(1,i)+1));
end

```

```

inputSlack1_G = (FVAL_G * DMU_G(1,1))- (SomaG1);

```

```

inputSlack2_G = (FVAL_G * DMU_G(2,1)) - (SomaG2);
outputSlack1_G = (FVAL_G * DMU_G(3,1)) - (SomaG3);
outputSlack2_G = (FVAL_G * DMU_G(4,1)) - (SomaG4);
outputSlack3_G = (FVAL_G * DMU_G(5,1)) - (SomaG5);

```

```

SomaH1=0;
SomaH2=0;
SomaH3=0;
SomaH4=0;
SomaH5=0;

```

```

for i=1:length(LambdaH_index)
    SomaH1 = SomaH1 + (lambdaH(1,i)) * (-DMU_H(1,LambdaH_index(1,i)+1));
    single(SomaH1);
end
for i=1:length(LambdaH_index)
    SomaH2 = SomaH2 + (lambdaH(1,i)) * (-DMU_H(2,LambdaH_index(1,i)+1));
    single(SomaH2);
end
for i=1:length(LambdaH_index)
    SomaH3 = SomaH3 + (lambdaH(1,i)) * (-DMU_H(3,LambdaH_index(1,i)+1));
end
for i=1:length(LambdaH_index)
    SomaH4 = SomaH4 + (lambdaH(1,i)) * (-DMU_H(4,LambdaH_index(1,i)+1));
end
for i=1:length(LambdaH_index)
    SomaH5 = SomaH5 + (lambdaH(1,i)) * (-DMU_H(5,LambdaH_index(1,i)+1));
end

```

```

inputSlack1_H = (FVAL_H * DMU_H(1,1)) - (SomaH1);
inputSlack2_H = (FVAL_H * DMU_H(2,1)) - (SomaH2);
outputSlack1_H = (FVAL_H * DMU_H(3,1)) - (SomaH3);
outputSlack2_H = (FVAL_H * DMU_H(4,1)) - (SomaH4);
outputSlack3_H = (FVAL_H * DMU_H(5,1)) - (SomaH5);

```

```

SomaI1=0;
SomaI2=0;
SomaI3=0;
SomaI4=0;
SomaI5=0;

```

```

for i=1:length(LambdaI_index)
    SomaI1 = SomaI1 + (lambdaI(1,i)) * (-DMU_I(1,LambdaI_index(1,i)+1));
    single(SomaI1);
end
for i=1:length(LambdaI_index)
    SomaI2 = SomaI2 + (lambdaI(1,i)) * (-DMU_I(2,LambdaI_index(1,i)+1));
    single(SomaI2);
end
for i=1:length(LambdaI_index)
    SomaI3 = SomaI3 + (lambdaI(1,i)) * (-DMU_I(3,LambdaI_index(1,i)+1));
end
for i=1:length(LambdaI_index)
    SomaI4 = SomaI4 + (lambdaI(1,i)) * (-DMU_I(4,LambdaI_index(1,i)+1));
end
for i=1:length(LambdaI_index)
    SomaI5 = SomaI5 + (lambdaI(1,i)) * (-DMU_I(5,LambdaI_index(1,i)+1));
end

```

```

inputSlack1_I = (FVAL_I * DMU_I(1,1)) - (SomaI1);
inputSlack2_I = (FVAL_I * DMU_I(2,1)) - (SomaI2);
outputSlack1_I = (FVAL_I * DMU_I(3,1)) - (SomaI3);
outputSlack2_I = (FVAL_I * DMU_I(4,1)) - (SomaI4);
outputSlack3_I = (FVAL_I * DMU_I(5,1)) - (SomaI5);

```

```

[lambda_A,FVAL_A,exitflag,output,lambda]=linprog(f,-DMU_A,-b_A,[],[],lb);

[lambda_B,FVAL_B,exitflag,output,lambda]=linprog(f,-DMU_B,-b_B,[],[],lb);

[lambda_C,FVAL_C,exitflag,output,lambda]=linprog(f,-DMU_C,-b_C,[],[],lb);

[lambda_D,FVAL_D,exitflag,output,lambda]=linprog(f,-DMU_D,-b_D,[],[],lb);

[lambda_E,FVAL_E,exitflag,output,lambda]=linprog(f,-DMU_E,-b_E,[],[],lb);

[lambda_F,FVAL_F,exitflag,output,lambda]=linprog(f,-DMU_F,-b_F,[],[],lb);

[lambda_G,FVAL_G,exitflag,output,lambda]=linprog(f,-DMU_G,-b_G,[],[],lb);

[lambda_H,FVAL_H,exitflag,output,lambda]=linprog(f,-DMU_H,-b_H,[],[],lb);

[lambda_I,FVAL_I,exitflag,output,lambda]=linprog(f,-DMU_I,-b_I,[],[],lb);

```

% Targets of the inputs from DMU

```

if (FVAL_A~=1)
    Radial1_A=FVAL_A*DMU_A(1,1);
    Radial2_A=FVAL_A*DMU_A(2,1);
    Radial3_A=FVAL_A*DMU_A(3,2);
    Radial4_A=FVAL_A*DMU_A(4,2);
    Radial5_A=FVAL_A*DMU_A(5,2);
end

if (FVAL_B~=1)
    Radial1_B=FVAL_B*DMU_B(1,1);
    Radial2_B=FVAL_B*DMU_B(2,1);
    Radial3_B=FVAL_B*DMU_B(3,3);
    Radial4_B=FVAL_B*DMU_B(4,3);
    Radial5_B=FVAL_B*DMU_B(5,3);
end

if (FVAL_C~=1)
    Radial1_C=FVAL_C*DMU_C(1,1);
    Radial2_C=FVAL_C*DMU_C(2,1);
    Radial3_C=FVAL_C*DMU_C(3,4);
    Radial4_C=FVAL_C*DMU_C(4,4);
    Radial5_C=FVAL_C*DMU_C(5,4);
end

if (FVAL_D~=1)
    Radial1_D=FVAL_D*DMU_D(1,1);
    Radial2_D=FVAL_D*DMU_D(2,1);

```



```

Radial3_D=FVAL_D*DMU_D(3,5);
Radial4_D=FVAL_D*DMU_D(4,5);
Radial5_D=FVAL_D*DMU_D(5,5);

end

if (FVAL_E~=1)
    Radial1_E=FVAL_E*DMU_E(1,1);
    Radial2_E=FVAL_E*DMU_E(2,1);
    Radial3_E=FVAL_E*DMU_E(3,6);
    Radial4_E=FVAL_E*DMU_E(4,6);
    Radial5_E=FVAL_E*DMU_E(5,6);

end

if (FVAL_F~=1)
    Radial1_F=FVAL_F*DMU_F(1,1);
    Radial2_F=FVAL_F*DMU_F(2,1);
    Radial3_F=FVAL_F*DMU_F(3,7);
    Radial4_F=FVAL_F*DMU_F(4,7);
    Radial5_F=FVAL_F*DMU_F(5,7);

end

if (FVAL_G~=1)
    Radial1_G=FVAL_G*DMU_G(1,1);
    Radial2_G=FVAL_G*DMU_G(2,1);
    Radial3_G=FVAL_G*DMU_G(3,8);
    Radial4_G=FVAL_G*DMU_G(4,8);
    Radial5_G=FVAL_G*DMU_G(5,8);

end

if (FVAL_H~=1)
    Radial1_H=FVAL_H*DMU_H(1,1);
    Radial2_H=FVAL_H*DMU_H(2,1);
    Radial3_H=FVAL_H*DMU_A(3,9);
    Radial4_H=FVAL_H*DMU_H(4,9);
    Radial5_H=FVAL_H*DMU_H(5,9);

end

if (FVAL_I~=1)
    Radial1_I=FVAL_I*DMU_I(1,1);
    Radial2_I=FVAL_I*DMU_I(2,1);
    Radial3_I=FVAL_I*DMU_I(3,10);
    Radial4_I=FVAL_I*DMU_A(4,10);
    Radial5_I=FVAL_I*DMU_A(5,10);

end

Target1_A = Radial1_A - inputSlack1_A;
Target2_A = Radial2_A - inputSlack2_A;
Target3_A = outputSlack1_A - DMU_A(3,2);
Target4_A = outputSlack2_A - DMU_A(4,2);
Target5_A = outputSlack3_A - DMU_A(5,2);

Target1_B = Radial1_B - inputSlack1_B;
Target2_B = Radial2_B - inputSlack2_B;

```

```

Target3_B = outputSlack1_B - DMU_A(3,3);
Target4_B = outputSlack2_B - DMU_A(4,3);
Target5_B = outputSlack3_B - DMU_A(5,3);

```

```

Target1_C = Radial1_C - inputSlack1_C;
Target2_C = Radial2_C - inputSlack2_C;
Target3_C = outputSlack1_C - DMU_A(3,4);
Target4_C = outputSlack2_C - DMU_A(4,4);
Target5_C = outputSlack3_C - DMU_A(5,4);

```

```

Target1_D = Radial1_D - inputSlack1_D;
Target2_D = Radial2_D - inputSlack2_D;
Target3_D = outputSlack1_D - DMU_A(3,5);
Target4_D = outputSlack2_D - DMU_A(4,5);
Target5_D = outputSlack3_D - DMU_A(5,5);

```

```

Target1_E = Radial1_E - inputSlack1_E;
Target2_E = Radial2_E - inputSlack2_E;
Target3_E = outputSlack1_E - DMU_A(3,6);
Target4_E = outputSlack2_E - DMU_A(4,6);
Target5_E = outputSlack3_E - DMU_A(5,6);

```

```

Target1_F = Radial1_F - inputSlack1_F;
Target2_F = Radial2_F - inputSlack2_F;
Target3_F = outputSlack1_F - DMU_A(3,7);
Target4_F = outputSlack2_F - DMU_A(4,7);
Target5_F = outputSlack3_F - DMU_A(5,7);

```

```

Target1_G = Radial1_G - inputSlack1_G;
Target2_G = Radial2_G - inputSlack2_G;
Target3_G = outputSlack1_G - DMU_A(3,8);
Target4_G = outputSlack2_G - DMU_A(4,8);
Target5_G = outputSlack3_G - DMU_A(5,8);

```

```

Target1_H = Radial1_H - inputSlack1_H;
Target2_H = Radial2_H - inputSlack2_H;
Target3_H = outputSlack1_H - DMU_A(3,9);
Target4_H = outputSlack2_H - DMU_A(4,9);
Target5_H = outputSlack3_H - DMU_A(5,9);

```

```

Target1_I = Radial1_I - inputSlack1_I;
Target2_I = Radial2_I - inputSlack2_I;
Target3_I = outputSlack1_I - DMU_A(3,10);
Target4_I = outputSlack2_I - DMU_A(4,10);
Target5_I = outputSlack3_I - DMU_A(5,10);

```

```

clc;

```

```

% Additive Model to Calculate Slacks

```

```

%DMU_Eq = [ 2 4 3 2 1 3 -1 0 0;
%          6 8 12 6 6 15 0 1 0;
%          14 16 12 4 3 3 0 0 1;
%          1 1 1 1 1 1 0 0 0;];
%A B C D E F G
DMU_Eq = [ 2 5 5 2 4 2 1 3 2 -1 0 0 0 0; % Output_1

```

```

2 50 30 125 20 30 18 50 30 0 -1 0 0 0; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 0 0 -1 0 0; % Output_3
163 2000 375 1210 230 220 190 300 205 0 0 0 1 0; % Input_1
215 233 220 245 223 235 240 230 235 0 0 0 1; % Input_2

```

```

objective = [ 0;0;0;0;0;0;0;0;1;1;1;1;1];

```

```

A_Eq = DMU_Eq(:,1);
B_Eq = DMU_Eq(:,2);
C_Eq = DMU_Eq(:,3);
D_Eq = DMU_Eq(:,4);
E_Eq = DMU_Eq(:,5);
F_Eq = DMU_Eq(:,6);
G_Eq = DMU_Eq(:,7);
H_Eq = DMU_Eq(:,8);
I_Eq = DMU_Eq(:,9);

```

```

lb2 = zeros(14,1);

```

```

[XA,SLACKA,EXITFLAG_SA,OUTPUT_A,lambdaXA]=linprog(-objective,[],[],DMU_Eq,A_Eq,lb2);
[XB,SLACKB,EXITFLAG_SB,OUTPUT_B,lambdaXB]=linprog(-objective,[],[],DMU_Eq,B_Eq,lb2);
[XC,SLACKC,EXITFLAG_SC,OUTPUT_C,lambdaXC]=linprog(-objective,[],[],DMU_Eq,C_Eq,lb2);
[XD,SLACKD,EXITFLAG_SD,OUTPUT_D,lambdaXD]=linprog(-objective,[],[],DMU_Eq,D_Eq,lb2);
[XE,SLACKE,EXITFLAG_SE,OUTPUT_E,lambdaXE]=linprog(-objective,[],[],DMU_Eq,E_Eq,lb2);
[XF,SLACKF,EXITFLAG_SF,OUTPUT_F,lambdaXF]=linprog(-objective,[],[],DMU_Eq,F_Eq,lb2);
options = optimset('LargeScale','off','Simplex','on','TolFun',1e-8,'MaxIter',10);
[XG,SLACKG,EXITFLAG_SG,OUTPUT_G,lambdaXG]=linprog(-
objective,[],[],DMU_Eq,G_Eq,lb2,[],[],options);
[XH,SLACKH,EXITFLAG_SH,OUTPUT_H,lambdaXH]=linprog(-
objective,[],[],DMU_Eq,H_Eq,lb2,[],[],options);
[XI,SLACKI,EXITFLAG_SI,OUTPUT_I,lambdaXI]=linprog(-
objective,[],[],DMU_Eq,I_Eq,lb2,[],[],options);

```

```

for a=1:14
    SL_A(a)=XA(a);
end

```

```

Slack1_A = SL_A(1,10);
Slack2_A = SL_A(1,11);
Slack3_A = SL_A(1,12);
Excess1_A = SL_A(1,13);
Excess2_A = SL_A(1,14);

```

```

for b=1:14
    SL_B(b)=XB(b);
end

```

```

Slack1_B = SL_B(1,10);
Slack2_B = SL_B(1,11);
Slack3_B = SL_B(1,12);
Excess1_B = SL_B(1,13);
Excess2_B = SL_B(1,14);

```

```

for c=1:14
    SL_C(c)=XC(c);
end

```

```

Slack1_C = SL_C(1,10);

```

```

    Slack2_C = SL_C(1,11);
    Slack3_C = SL_C(1,12);
    Excess1_C = SL_C(1,13);
    Excess2_C = SL_C(1,14);

for d=1:14
    SL_D(d)=XD(d);
end

    Slack1_D = SL_D(1,10);
    Slack2_D = SL_D(1,11);
    Slack3_D = SL_D(1,12);
    Excess1_D = SL_D(1,13);
    Excess2_D = SL_D(1,14);

for e=1:14
    SL_E(e)=XE(e);
end

    Slack1_E = SL_E(1,10);
    Slack2_E = SL_E(1,11);
    Slack3_E = SL_E(1,12);
    Excess1_E = SL_E(1,13);
    Excess2_E = SL_E(1,14);

for f=1:14
    SL_F(f)=XF(f);
end

    Slack1_F = SL_F(1,10);
    Slack2_F = SL_F(1,11);
    Slack3_F = SL_F(1,12);
    Excess1_F = SL_F(1,13);
    Excess2_F = SL_F(1,14);

for g=1:14
    SL_G(g)=XG(g);
end

    Slack1_G = SL_G(1,10);
    Slack2_G = SL_G(1,11);
    Slack3_G = SL_G(1,12);
    Excess1_G = SL_G(1,13);
    Excess2_G = SL_G(1,14);

for h=1:14
    SL_H(h)=XH(h);
end

    Slack1_H = SL_H(1,10);
    Slack2_H = SL_H(1,11);
    Slack3_H = SL_H(1,12);
    Excess1_H = SL_H(1,13);
    Excess2_H = SL_H(1,14);

for i=1:14
    SL_I(i)=XI(i);
end

```

```

Slack1_I = SL_I(1,10);
Slack2_I = SL_I(1,11);
Slack3_I = SL_I(1,12);
Excess1_I = SL_I(1,13);
Excess2_I = SL_I(1,14);

```

```
clc;
```

```
%Reports
```

```

disp('===== DEA
MATLAB Solver
=====');
disp('===== Eficiência =          RADIAL          = FOLGAS (SLACKS) ==
FALTAS    ==    ALVOS (TARGETS)    ==');
disp('== DMU == Theta = I1 == I2 == O1 == O2 == O3 = I1 == I2 == O1 == O2 ==
O3 == I1 == I2 == O1 == O2 == O3 ==');
disp('=====
=====
=====');
%Results summary from A
fprintf('== M.Net == %5.4f', FVAL_A);
fprintf(' ==');
fprintf(' %5.2f,Radial1_A);
fprintf(' ==');
fprintf(' %5.2f,Radial2_A);
fprintf(' ==');
fprintf(' %5.2f,Radial3_A);
fprintf(' ==');
fprintf(' %5.2f,Radial4_A);
fprintf(' ==');
fprintf(' %5.2f,Radial5_A);
fprintf(' ==');
fprintf(' %2.0f,inputSlack1_A);
fprintf(' ==');
fprintf(' %4.2f,inputSlack2_A);
fprintf(' ==');
fprintf(' %2.0f ',Target3_A);% this variable was change its semantic
fprintf('==');
fprintf(' %4.0f ',Target4_A);
fprintf(' ==');
fprintf(' %4.0f,Target5_A);
fprintf(' ==');
fprintf(' %4.0f ',Target1_A);
fprintf('==');
fprintf(' %5.1f,Target2_A);
fprintf(' ==');
fprintf(' %3.0f,outputSlack1_A);
fprintf(' ==');
fprintf(' %3.0f,outputSlack2_A);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_A);
fprintf(' ==');
fprintf('\n');
%Results summary from B
fprintf('== Baydenet == %5.4f', FVAL_B);
fprintf(' ==');
fprintf(' %5.2f,Radial1_B);
fprintf(' ==');
fprintf(' %5.2f,Radial2_B);

```

```

fprintf(' ==');
fprintf('%5.2f',Radial3_B);
fprintf(' ==');
fprintf('%5.0f',Radial4_B);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_B);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_B);
fprintf(' ==');
fprintf('%3.0f,inputSlack2_B);
fprintf(' == ');
fprintf('%2.0f,Target3_B);% this variable was change its semantic
fprintf(' ==');
fprintf(' %2.0f',Target4_B);
fprintf(' ==');
fprintf(' %2.0f',Target5_B);
fprintf(' ==');
fprintf(' %4.0f ',Target1_B);
fprintf(' ==');
fprintf(' %3.0f',Target2_B);
fprintf(' ==');
fprintf('%3.0f,outputSlack1_B);
fprintf(' ==');
fprintf(' %2.0f ',outputSlack2_B);
fprintf(' ==');
fprintf(' %4.0f,outputSlack3_B);
fprintf(' ==');
fprintf('\n');
%Results summary from C
fprintf('== ADLLink == %5.4f, FVAL_C);
fprintf(' ==');
fprintf(' %5.2f',Radial1_C);
fprintf(' ==');
fprintf(' %5.2f',Radial2_C);
fprintf(' ==');
fprintf(' %5.2f',Radial3_C);
fprintf(' ==');
fprintf(' %5.0f',Radial4_C);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_C);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_C);
fprintf(' ==');
fprintf('%3.0f,inputSlack2_C);
fprintf(' ==');
fprintf('%3.0f,Target3_C);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f',Target4_C);
fprintf(' ==');
fprintf('%3.0f,Target5_C);
fprintf(' ==');
fprintf(' %3.0f ',Target1_C);
fprintf(' ==');
fprintf(' %3.0f ',Target2_C);
fprintf(' ==');
fprintf('%3.0f ',outputSlack1_C);
fprintf(' ==');
fprintf(' %2.0f ',outputSlack2_C);
fprintf(' ==');
fprintf(' %5.0f ',outputSlack3_C);

```

```

fprintf(' ==');
fprintf('\n');
%Results summary from D
fprintf('== NetLine == %5.4f,    FVAL_D);
fprintf(' ==');
fprintf('%3.2f,Radial1_D);
fprintf(' ==');
fprintf(' %5.2f,Radial2_D);
fprintf(' ==');
fprintf(' %4.2f,Radial3_D);
fprintf(' ==');
fprintf(' %3.0f,Radial4_D);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_D);
fprintf('==');
fprintf('%2.0f,inputSlack1_D);
fprintf(' ==');
fprintf(' %2.0f ',inputSlack2_D);
fprintf(' ==');
fprintf('%3.0f,Target3_D);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f,Target4_D);
fprintf(' ==');
fprintf(' %2.0f,Target5_D);
fprintf(' ==');
fprintf('%3.0f ',Target1_D);
fprintf('==');
fprintf(' %3.0f ',Target2_D);
fprintf('==');
fprintf(' %2.0f ',outputSlack1_D);
fprintf('==');
fprintf('%2.0f,outputSlack2_D);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_D);
fprintf(' ==');
fprintf('\n');

```

```

%Results summary from E
fprintf('== CEPlug == %5.4f,    FVAL_E);
fprintf(' ==');
fprintf(' %5.2f,Radial1_E);
fprintf(' ==');
fprintf(' %5.2f,Radial2_E);
fprintf(' ==');
fprintf(' %4.2f,Radial3_E);
fprintf(' == ');
fprintf('%4.0f,Radial4_E);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_E);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_E);
fprintf(' ==');
fprintf(' %2.0f,inputSlack2_E);
fprintf(' ==');
fprintf('%3.0f,Target3_E);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %2.0f,Target4_E);
fprintf(' ==');
fprintf(' %2.0f ',Target5_E);
fprintf(' ==');

```

```

fprintf(' %2.0f ',Target1_E);
fprintf('==');
fprintf(' %3.0f ',Target2_E);
fprintf('==');
fprintf('%3.0f,outputSlack1_E);
fprintf(' ==');
fprintf('%3.0f,outputSlack2_E);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_E);
fprintf(' ==');
fprintf('\n');%Results summary from F
fprintf('== HotLink == %5.4f', FVAL_F);
fprintf(' ==');
fprintf(' %5.2f',Radial1_F);
fprintf(' ==');
fprintf(' %5.2f',Radial2_F);
fprintf(' ==');
fprintf('%5.2f,Radial3_F);
fprintf(' ==');
fprintf('%3.2f,Radial4_F);
fprintf(' ==');
fprintf('%5.2f,Radial5_F);
fprintf('==');
fprintf('%2.0f,inputSlack1_F);
fprintf(' ==');
fprintf(' %5.2f,inputSlack2_F);
fprintf(' ==');
fprintf('%3.0f,Target3_F);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f,Target4_F);
fprintf(' ==');
fprintf('%5.2f,Target5_F);
fprintf('==');
fprintf('%4.2f,Target1_F);
fprintf('==');
fprintf('%5.2f ',Target2_F);
fprintf('==');
fprintf('%3.0f,outputSlack1_F);
fprintf(' ==');
fprintf('%3.0f,outputSlack2_F);
fprintf(' ==');
fprintf(' %5.2f,outputSlack3_F);
fprintf(' ==');
fprintf('\n');

```

```

%Results summary from G
fprintf('== Cz.Net == %5.4f', FVAL_G);
fprintf(' ==');
fprintf(' %5.2f,Radial1_G);
fprintf(' ==');
fprintf(' %3.2f ',Radial2_G);
fprintf('==');
fprintf(' %3.2f,Radial3_G);
fprintf(' ==');
fprintf('%3.2f ',Radial4_G);
fprintf('==');
fprintf('%5.2f,Radial5_G);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_G);
fprintf(' ==');

```



```

fprintf(' %5.2f',inputSlack2_G);
fprintf(' ==');
fprintf('%3.2f',Target3_G);% this variable was change its semantic wick outputSlack
fprintf('==');
fprintf(' %2.0f ',Target4_G);
fprintf('==');
fprintf(' %4.0f',Target5_G);
fprintf(' ==');
fprintf(' %3.0f',Target1_G);
fprintf(' ==');
fprintf(' %5.2f',Target2_G);
fprintf(' ==');
fprintf('%3.2f,outputSlack1_G);
fprintf('==');
fprintf('%3.0f,outputSlack2_G);
fprintf(' ==');
fprintf(' %5.0f',outputSlack3_G);
fprintf(' ==');
fprintf('\n');
%Results summary from H
fprintf('== WiNet == %5.4f, FVAL_H);
fprintf(' ==');
fprintf(' %5.2f',Radial1_H);
fprintf(' ==');
fprintf(' %5.2f',Radial2_H);
fprintf(' ==');
fprintf(' %4.2f',Radial3_H);
fprintf(' ==');
fprintf(' %2.0f',Radial4_H);
fprintf(' ==');
fprintf(' %5.0f',Radial5_H);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_H);
fprintf(' ==');
fprintf(' %2.0f',inputSlack2_H);
fprintf(' ==');
fprintf(' %2.0f ',Target3_H);% this variable was change its semantic wick outputSlack
fprintf('==');
fprintf(' %2.0f',Target4_H);
fprintf(' ==');
fprintf(' %2.0f',Target5_H);
fprintf(' ==');
fprintf(' %3.0f',Target1_H);
fprintf(' ==');
fprintf(' %3.0f ',Target2_H);
fprintf('==');
fprintf('%3.0f,outputSlack1_H);
fprintf(' ==');
fprintf(' %2.0f',outputSlack2_H);
fprintf(' ==');
fprintf(' %5.0f',outputSlack3_H);
fprintf(' ==');
fprintf('\n');%Results summary from I
fprintf('= Throughput = %5.4f, FVAL_I);
fprintf(' ==');
fprintf(' %5.2f',Radial1_I);
fprintf(' ==');
fprintf(' %5.2f',Radial2_I);
fprintf(' ==');
fprintf('%5.2f',Radial3_I);

```

```

fprintf('==');
fprintf('%5.2f,Radial4_I);
fprintf('==');
fprintf('%5.2f,Radial5_I);
fprintf('==');
fprintf('%2.0f,inputSlack1_I);
fprintf('==');
fprintf(' %5.2f,inputSlack2_I);
fprintf('==');
fprintf(' %2.0f,Target3_I);% this variable was change its semantic wich outputSlack
fprintf('==');
fprintf(' %2.0f,Target4_I);
fprintf('==');
fprintf('%5.2f,Target5_I);
fprintf('==');
fprintf('%5.2f,Target1_I);
fprintf('==');
fprintf('%5.2f ',Target2_I);
fprintf('==');
fprintf(' %2.0f,outputSlack1_I);
fprintf('==');
fprintf(' %2.0f,outputSlack2_I);
fprintf('==');
fprintf(' %5.0f,outputSlack3_I);
fprintf('==');
fprintf('\n');
disp('=====
=====
=====');
fprintf('\n');

% MBF Reports
disp('=====
=====');
disp('===== DEA MATLAB Solver - MBF
=====');
disp('===== SOMA (FOLGAS + FALTAS) == FOLGAS (INPUT) == FALTAS
(OUTPUT) ==');
disp('== DMU == I1 == I2 == O1 == O2 == O3 ==');
disp('=====
=====');
fprintf('== M.Net ==');
fprintf(' %5.2f,-SLACKA);
fprintf('==');
fprintf(' %5.2f,Excess1_A);
fprintf('==');
fprintf(' %5.2f, Excess2_A);
fprintf('==');
fprintf(' %5.2f,Slack1_A);
fprintf('==');
fprintf(' %5.2f,Slack2_A);
fprintf('==');
fprintf(' %5.2f,Slack3_A);
fprintf('==');
fprintf('\n');
fprintf('== Baydenet ==');
fprintf(' %5.2f ',-SLACKB);
fprintf('==');
fprintf(' %5.2f,Excess1_B);
fprintf('==');

```

```

fprintf(' %5.2f, Excess2_B);
fprintf(' ==');
fprintf(' %5.2f,Slack1_B);
fprintf(' ==');
fprintf(' %5.2f,Slack2_B);
fprintf(' ==');
fprintf(' %5.2f,Slack3_B);
fprintf(' ==');
fprintf('\n');
fprintf('== ADLLink ==');
fprintf(' %5.2f ',-SLACKC);
fprintf(' ==');
fprintf(' %5.2f,Excess1_C);
fprintf(' ==');
fprintf(' %5.2f, Excess2_C);
fprintf(' ==');
fprintf(' %5.2f,Slack1_C);
fprintf(' ==');
fprintf(' %5.2f,Slack2_C);
fprintf(' ==');
fprintf(' %5.2f,Slack3_C);
fprintf(' ==');
fprintf('\n');
fprintf('== NetLine ==');
fprintf(' %5.2f ',-SLACKD);
fprintf(' ==');
fprintf(' %5.2f,Excess1_D);
fprintf(' ==');
fprintf(' %5.2f, Excess2_D);
fprintf(' ==');
fprintf(' %5.2f,Slack1_D);
fprintf(' ==');
fprintf(' %5.2f,Slack2_D);
fprintf(' ==');
fprintf(' %5.2f,Slack3_D);
fprintf(' ==');
fprintf('\n');
fprintf('== CEPlug ==');
fprintf(' %5.2f ',-SLACKE);
fprintf(' ==');
fprintf(' %5.2f,Excess1_E);
fprintf(' ==');
fprintf(' %5.2f, Excess2_E);
fprintf(' ==');
fprintf(' %5.2f,Slack1_E);
fprintf(' ==');
fprintf(' %5.2f,Slack2_E);
fprintf(' ==');
fprintf(' %5.2f,Slack3_E);
fprintf(' ==');
fprintf('\n');
fprintf('== HotLink ==');
fprintf(' %5.2f',-SLACKF);
fprintf(' ==');
fprintf(' %5.2f,Excess1_F);
fprintf(' ==');
fprintf(' %5.2f, Excess2_F);
fprintf(' ==');
fprintf(' %5.2f,Slack1_F);
fprintf(' ==');

```

```

fprintf(' %5.2f,Slack2_F);
fprintf(' ==');
fprintf(' %5.2f,Slack3_F);
fprintf(' ==');
fprintf('\n');
fprintf('== Cz.Net ==');
fprintf(' %5.2f,-SLACKG);
fprintf(' ==');
fprintf(' %5.2f,Excess1_G);
fprintf(' ==');
fprintf(' %5.2f, Excess2_G);
fprintf(' ==');
fprintf(' %5.2f,Slack1_G);
fprintf(' ==');
fprintf(' %5.2f,Slack2_G);
fprintf(' ==');
fprintf(' %5.2f,Slack3_G);
fprintf(' ==');
fprintf('\n');
fprintf('== WiNet ==');
fprintf(' %5.2f ',-SLACKH);
fprintf(' ==');
fprintf(' %5.2f,Excess1_H);
fprintf(' ==');
fprintf(' %5.2f, Excess2_H);
fprintf(' ==');
fprintf(' %5.2f,Slack1_H);
fprintf(' ==');
fprintf(' %5.2f,Slack2_H);
fprintf(' ==');
fprintf(' %5.2f,Slack3_H);
fprintf(' ==');
fprintf('\n');
fprintf('== Throughput ==');
fprintf(' %5.2f,-SLACKI);
fprintf(' ==');
fprintf(' %5.2f,Excess1_I);
fprintf(' ==');
fprintf(' %5.2f, Excess2_I);
fprintf(' ==');
fprintf(' %5.2f,Slack1_I);
fprintf(' ==');
fprintf(' %5.2f,Slack2_I);
fprintf(' ==');
fprintf(' %5.2f,Slack3_I);
fprintf(' ==');
fprintf('\n');
disp('=====
=====');

```

% Peers and Benchmarks Reports

```

disp('=====
=====');
disp('== DMU == Parceiro(s) de Excelência == Lambdas ==');
disp('=====
=====');
fprintf('== 1 - M.Net ==');
fprintf(' %5.2f,LambdaA_index);
fprintf(' ==');

```

```

fprintf(' %5.2f',lambdaA);
fprintf('\n');
fprintf('== 2 - Baydenet ==');
fprintf(' %5.2f',LambdaB_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaB);
fprintf('\n');
fprintf('== 3 - ADLLink ==');
fprintf(' %5.2f',LambdaC_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaC);
fprintf('\n');
fprintf('== 4 - NetLine ==');
fprintf(' %5.2f',LambdaD_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaD);
fprintf('\n');
fprintf('== 5 - CEPlug ==');
fprintf(' %5.2f',LambdaE_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaE);
fprintf('\n');
fprintf('== 6 - HotLink ==');
fprintf(' %5.2f',LambdaF_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaF);
fprintf('\n');
fprintf('== 7 - CzNet ==');
fprintf(' %5.2f',LambdaG_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaG);
fprintf('\n');
fprintf('== 8 - WiNet ==');
fprintf(' %5.2f',LambdaH_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaH);
fprintf('\n');
fprintf('== 9 - Throughput ==');
fprintf(' %5.2f',LambdaI_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaI);
fprintf('\n');
disp('=====');
=====);

```

```

% Graphics
x_axes = [FVAL_A,FVAL_B,FVAL_C,FVAL_D,FVAL_E,FVAL_F,FVAL_G,FVAL_H,FVAL_I];
figure1=figure;
barh(x_axes(1,1:9), 'DisplayName', {'A','B','C','D','E','F','G','H','T'});
title('Eficiência das DMUs');
set(gca,'YTickLabel', {'A','B','C','D','E','F','G','H','T'})

```

% DEA - BCC - RVE - Multiplier - Input Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;  
clear;  
b = zeros(9,1); % constraints
```

```
DMU= [2 2 1400 -163 -215 +1;  
      5 50 50000 -2000 -233 +1;  
      5 30 10000 -375 -220 +1;  
      2 125 40000 -1210 -245 +1;  
      4 20 15000 -230 -223 +1;  
      2 30 20000 -220 -235 +1;  
      1 18 4000 -190 -240 +1;  
      3 50 35000 -300 -230 +1;  
      2 30 20000 -205 -235 +1;];
```

```
f_A = [2 2 1400 0 0 1]; % objective function  
f_B = [5 50 50000 0 0 1];  
f_C = [5 30 10000 0 0 1];  
f_D = [2 125 40000 0 0 1];  
f_E = [4 20 15000 0 0 1];  
f_F = [2 30 20000 0 0 1];  
f_G = [1 18 4000 0 0 1];  
f_H = [3 50 35000 0 0 1];  
f_I = [2 30 20000 0 0 1];  
DMUA_Aeq = [0 0 0 163 215 0];  
DMUB_Aeq = [0 0 0 2000 233 0];  
DMUC_Aeq = [0 0 0 375 220 0];  
DMUD_Aeq = [0 0 0 1210 245 0];  
DMUE_Aeq = [0 0 0 230 223 0];  
DMUF_Aeq = [0 0 0 220 235 0];  
DMUG_Aeq = [0 0 0 190 240 0];  
DMUH_Aeq = [0 0 0 300 230 0];  
DMUI_Aeq = [0 0 0 205 235 0];
```

```
Beq = 1;
```

```
lb = zeros(6,1);  
clc;
```

% Calculation of Phi

```
[XA,FVAL_A,EXITFLAG,OUTPUT]=linprog(-f_A,DMU,b,DMUA_Aeq,Beq,lb);  
[XB,FVAL_B,EXITFLAG,OUTPUT]=linprog(-f_B,DMU,b,DMUB_Aeq,Beq,lb);  
[XC,FVAL_C,EXITFLAG,OUTPUT]=linprog(-f_C,DMU,b,DMUC_Aeq,Beq,lb);  
[XD,FVAL_D,EXITFLAG,OUTPUT]=linprog(-f_D,DMU,b,DMUD_Aeq,Beq,lb);  
[XE,FVAL_E,EXITFLAG,OUTPUT]=linprog(-f_E,DMU,b,DMUE_Aeq,Beq,lb);  
[XF,FVAL_F,EXITFLAG,OUTPUT]=linprog(-f_F,DMU,b,DMUF_Aeq,Beq,lb);  
[XG,FVAL_G,EXITFLAG,OUTPUT]=linprog(-f_G,DMU,b,DMUG_Aeq,Beq,lb);  
[XH,FVAL_H,EXITFLAG,OUTPUT]=linprog(-f_H,DMU,b,DMUH_Aeq,Beq,lb);
```

```

[XI,FVAL_I,EXITFLAG,OUTPUT]=linprog(-f_I,DMU,b,DMUI_Aeq,Beq,lb);

% BCC/ M / I

f = [0 0 0 0 0 0 0 0 0 1];
b_A = [-163 -215 0 0 0];

DMU_A = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -2; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -1400;]; % Output_3

b_B = [-2000 -233 0 0 0];

DMU_B = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -50000;]; % Output_3

b_C = [-375 -220 0 0 0];

DMU_C = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -10000;]; % Output_3

b_D = [-1210 -245 0 0 0];

DMU_D = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -125; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -40000;]; % Output_3

b_E = [-230 -223 0 0 0];

DMU_E = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -4; % Output_1
2 50 30 125 20 30 18 50 30 -20; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -15000;]; % Output_3

b_F = [-220 -235 0 0 0];

DMU_F = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

b_G = [-190 -240 0 0 0];

DMU_G = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -1; % Output_1
2 50 30 125 20 30 18 50 30 -18; % Output_2

```

```

1400 50000 10000 40000 15000 20000 4000 35000 20000 -4000;]; % Output_3

b_H = [ -300 -230 0 0 0];

DMU_H = [ -163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -3; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -35000;]; % Output_3

b_I = [ -205 -235 0 0 0];

DMU_I = [ -163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

lb= zeros(9,1);
% Technical Efficiency CCR
[lambda_A,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f,-DMU_A,-b_A,[],[],lb);
CCR_A=-1/FVAL_1;

[lambda_B,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f,-DMU_B,-b_B,[],[],lb);
CCR_B=-1/FVAL_2;

[lambda_C,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f,-DMU_C,-b_C,[],[],lb);
CCR_C=-1/FVAL_3;

[lambda_D,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f,-DMU_D,-b_D,[],[],lb);
CCR_D=-1/FVAL_4;

[lambda_E,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f,-DMU_E,-b_E,[],[],lb);
CCR_E=-1/FVAL_5;

[lambda_F,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f,-DMU_F,-b_F,[],[],lb);
CCR_F=-1/FVAL_6;

[lambda_G,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f,-DMU_G,-b_G,[],[],lb);
CCR_G=-1/FVAL_7;

[lambda_H,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f,-DMU_H,-b_H,[],[],lb);
CCR_H=-1/FVAL_8;

[lambda_I,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f,-DMU_I,-b_I,[],[],lb);
CCR_I=-1/FVAL_9;

```

% Technical Efficiency versus Scale Efficiency - Reports

```

clc;

disp('=====');
);
disp('          Efficiencies');

disp('=====');
);
disp('DMU      -   CCR      -   BCC      ');

```



```

disp('=====')
);
fprintf('== A ==      %5.4f,CCR_A);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_A);
fprintf('\n');
fprintf('== B ==      %5.4f,CCR_B);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_B);
fprintf('\n');
fprintf('== C ==      %5.4f,CCR_C);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_C);
fprintf('\n');
fprintf('== D ==      %5.4f,CCR_D);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_D);
fprintf('\n');
fprintf('== E ==      %5.4f,CCR_E);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_E);
fprintf('\n');
fprintf('== F ==      %5.4f,CCR_F);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_F);
fprintf('\n');
fprintf('== G ==      %5.4f,CCR_G);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_G);
fprintf('\n');
fprintf('== H ==      %5.4f,CCR_H);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_H);
fprintf('\n');
fprintf('== I ==      %5.4f,CCR_I);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_I);
fprintf('\n');

disp('=====')
);

```

% BCC – Graphic

```

x_axes = [-FVAL_A,-FVAL_B,-FVAL_C,-FVAL_D,-FVAL_E,-FVAL_F,-FVAL_G,-FVAL_H,-
FVAL_I];
figure1=figure;
barh(x_axes(1,1:9), 'DisplayName',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'});
title('Eficiência das DMUs - BCC');
set(gca,'YTickLabel',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'})

```

% CCR – Graphic

```

X_axes = [CCR_A,CCR_B,CCR_C,CCR_D,CCR_E,CCR_F,CCR_G,CCR_H,CCR_I];
figure2=figure;
barh(X_axes(1,1:9), 'DisplayName',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'});

```

```
title('Eficiência das DMUs - CCR');  
set(gca,'YTickLabel',  
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'})
```

% CCR - Envelopment – Output – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

clear;

clc;

f = [0 0 0 0 0 0 0 0 1];

b_A = [-163 -215 0 0 0];

DMU_A = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -2; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -1400;]; % Output_3

b_B = [-2000 -233 0 0 0];

DMU_B = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -50000;]; % Output_3

b_C = [-375 -220 0 0 0];

DMU_C = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -10000;]; % Output_3

b_D = [-1210 -245 0 0 0];

DMU_D = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -125; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -40000;]; % Output_3

b_E = [-230 -223 0 0 0];

DMU_E = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -4; % Output_1
2 50 30 125 20 30 18 50 30 -20; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -15000;]; % Output_3

b_F = [-220 -235 0 0 0];

DMU_F = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

b_G = [-190 -240 0 0 0];

DMU_G = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -1; % Output_1

```

2 50 30 125 20 30 18 50 30 -18; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -4000;]; % Output_3

b_H = [-300 -230 0 0 0];

DMU_H = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -3; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -35000;]; % Output_3

b_I = [-205 -235 0 0 0];

DMU_I = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

lb= zeros(9,1);

% Calculation of Technical Efficiency
[XA,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f,-DMU_A,-b_A,[],[],lb);
disp('A eficácia da DMU_A é:');
disp(-1/FVAL_1);

[XB,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f,-DMU_B,-b_B,[],[],lb);
disp('A eficácia da DMU_B é:');
disp(-1/FVAL_2);

[XC,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f,-DMU_C,-b_C,[],[],lb);
disp('A eficácia da DMU_C é:');
disp(-1/FVAL_3);

[XD,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f,-DMU_D,-b_D,[],[],lb);
disp('A eficácia da DMU_D é:');
disp(-1/FVAL_4);

[XE,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f,-DMU_E,-b_E,[],[],lb);
disp('A eficácia da DMU_E é:');
disp(-1/FVAL_5);

[XF,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f,-DMU_F,-b_F,[],[],lb);
disp('A eficácia da DMU_F é:');
disp(-1/FVAL_6);

[XG,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f,-DMU_G,-b_G,[],[],lb);
disp('A eficácia da DMU_G é:');
disp(-1/FVAL_7);

[XH,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f,-DMU_H,-b_H,[],[],lb);
disp('A eficácia da DMU_H é:');
disp(-1/FVAL_8);

[XI,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f,-DMU_I,-b_I,[],[],lb);
disp('A eficácia da DMU_I é:');
disp(-1/FVAL_9);

```

% DEA - CCR - Multipliers Mode - Input Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;  
clear;  
b = [ 0; 0; 0; 0; 0; 0; 0; 0; 0;]; % constraints
```

```
Beq = 1;
```

```
DMU= [2 2 1400 -163 -215;  
      5 50 50000 -2000 -233;  
      5 30 10000 -375 -220;  
      2 125 40000 -1210 -245;  
      4 20 15000 -230 -223;  
      2 30 20000 -220 -235;  
      1 18 4000 -190 -240;  
      3 50 35000 -300 -230;  
      2 30 20000 -205 -235;];
```

```
f_A = [2 2 1400 0 0]; % objective function
```

```
DMUA_Aeq = [0 0 0 163 215;];
```

```
f_B = [5 50 50000 0 0];
```

```
DMUB_Aeq = [0 0 0 2000 233];
```

```
f_C = [ 5; 30; 10000; 0; 0;];
```

```
DMUC_Aeq = [ 0 0 0 375 220;];
```

```
f_D = [2;125; 40000; 0;0;];
```

```
DMUD_Aeq=[ 0 0 0 1210 245;];
```

```
f_E = [4; 20; 15000; 0; 0;];
```

```
DMUE_Aeq=[ 0 0 0 230 223;];
```

```
f_F = [2; 30; 20000; 0; 0;];
```

```
DMUF_Aeq = [ 0 0 0 220 235;];
```

```
f_G = [1 18 4000 0 0;];
```

```
DMUG_Aeq = [ 0 0 0 190 240];
```

```
f_H = [ 3; 50; 35000; 0; 0;];
```

```
DMUH_Aeq = [ 0 0 0 300 230;];
```

```
f_I = [ 2; 30; 20000; 0; 0;];
```

```
DMUI_Aeq = [ 0 0 0 205 235;];
```

```
lb =zeros(5,1);
```

```
% Calculation of Technical Efficiency
```

```
[XA,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f_A,DMU,b,DMUA_Aeq,Beq,lb);
```

```
disp('A eficácia da DMU_A é:');  
disp(-FVAL_1);
```

```
[XB,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f_B,DMU,b,DMUB_Aeq,Beq,lb);  
disp('A eficácia da DMU_B é:');  
disp(-FVAL_2);
```

```
[XC,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f_C,DMU,b,DMUC_Aeq,Beq,lb);  
disp('A eficácia da DMU_C é:');  
disp(-FVAL_3);
```

```
[XD,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f_D,DMU,b,DMUD_Aeq,Beq,lb);  
disp('A eficácia da DMU_D é:');  
disp(-FVAL_4);
```

```
[XE,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f_E,DMU,b,DMUE_Aeq,Beq,lb);  
disp('A eficácia da DMU_E é:');  
disp(-FVAL_5);
```

```
[XF,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f_F,DMU,b,DMUF_Aeq,Beq,lb);  
disp('A eficácia da DMU_F é:');  
disp(-FVAL_6);
```

```
[XG,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f_G,DMU,b,DMUG_Aeq,Beq,lb);  
disp('A eficácia da DMU_G é:');  
disp(-FVAL_7);
```

```
[XH,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f_H,DMU,b,DMUH_Aeq,Beq,lb);  
disp('A eficácia da DMU_H é:');  
disp(-FVAL_8);
```

```
[XI,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f_I,DMU,b,DMUI_Aeq,Beq,lb);  
disp('A eficácia da DMU_I é:');  
disp(-FVAL_9);
```

```
theta=[-FVAL_1 -FVAL_2 -FVAL_3 -FVAL_4 -FVAL_5 -FVAL_6 -FVAL_7 -FVAL_8 -FVAL_9];  
theta_A=-FVAL_1;  
theta_B=-FVAL_2;  
theta_C=-FVAL_3;  
theta_D=-FVAL_4;  
theta_E=-FVAL_5;  
theta_F=-FVAL_6;  
theta_G=-FVAL_7;  
theta_H=-FVAL_8;  
theta_I=-FVAL_9;
```

% Slacks

```
if (theta_A||1)  
    Input1=DMU(1,4);  
    Input2=DMU(1,5);  
    disp('Valores imutáveis da entrada da DMU_A:');  
    disp('Input1:' +Input1);  
    disp('Input2:' +Input2);  
end  
if (theta_A||1)  
    Input1=DMU(1,4)*theta_A;  
    Input2=DMU(1,5)*theta_A;  
    disp('Novos valores de entrada da DMU_A:');
```

```

    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_B||1)
    Input1=DMU(2,4);
    Input2=DMU(2,5);
    disp('Valores imutáveis da entrada da DMU_B:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_B||1)
    Input1=DMU(2,4)*theta_B;
    Input2=DMU(2,5)*theta_B;
    disp('Novos valores de entrada da DMU_B:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_C||1)
    Input1=DMU(3,4);
    Input2=DMU(3,5);
    disp('Valores imutáveis da entrada da DMU_C:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_C||1)
    Input1=DMU(2,4)*theta_C;
    Input2=DMU(2,5)*theta_C;
    disp('Novos valores de entrada da DMU_C:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_D||1)
    Input1=DMU(4,4);
    Input2=DMU(4,5);
    disp('Valores imutáveis da entrada da DMU_D:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_D||1)
    Input1=DMU(4,4)*theta_D;
    Input2=DMU(4,5)*theta_D;
    disp('Novos valores de entrada da DMU_D:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end

```

```

if (theta_E||1)
    Input1=DMU(5,4);
    Input2=DMU(5,5);
    disp('Valores imutáveis da entrada da DMU_E:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_E||1)
    Input1=DMU(5,4)*theta_E;
    Input2=DMU(5,5)*theta_E;
    disp('Novos valores de entrada da DMU_E:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_F||1)
    Input1=DMU(6,4);
    Input2=DMU(6,5);
    disp('Valores imutáveis da entrada da DMU_F:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_F||1)
    Input1=DMU(6,4)*theta_F;
    Input2=DMU(6,5)*theta_F;
    disp('Novos valores de entrada da DMU_F:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_G||1)
    Input1=DMU(7,4);
    Input2=DMU(7,5);
    disp('Valores imutáveis da entrada da DMU_G:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_G||1)
    Input1=DMU(7,4)*theta_G;
    Input2=DMU(7,5)*theta_G;
    disp('Novos valores de entrada da DMU_G:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end

if (theta_H||1)
    Input1=DMU(8,4);
    Input2=DMU(8,5);
    disp('Valores imutáveis da entrada da DMU_H:');
    disp('Input1:');

```



```

    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_H||1)
    Input1=DMU(8,4)*theta_H;
    Input2=DMU(8,5)*theta_H;
    disp('Novos valores de entrada da DMU_H:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end

if (theta_I||1)
    Input1=DMU(9,4);
    Input2=DMU(9,5);
    disp('Valores imutáveis da entrada da DMU_I:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_I||1)
    Input1=DMU(9,4)*theta_I;
    Input2=DMU(9,5)*theta_I;
    disp('Novos valores de entrada da DMU_I:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
end

```

% DEA - CCR - Multiplier - Output Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;
b = [ 1; 0; 0; 0; 0; 0; 0; 0; 0;]; % constraints

%DMU_A = [ 2 0 0; 2 -6 -14; 4 -8 -16; 3 -12 -12; 2 -6 -4; 1 -6 -3; 3 -15 -3];
DMU= [2 2 1400 -163 -215;
      5 50 50000 -2000 -233;
      5 30 10000 -375 -220;
      2 125 40000 -1210 -245;
      4 20 15000 -230 -223;
      2 30 20000 -220 -235;
      1 18 4000 -190 -240;
      3 50 35000 -300 -230;
      2 30 20000 -205 -235;];
f_A = [0; 0; 0; 163; 215;]; % objective function

DMUA_Aeq = [2 2 1400 0 0]
DMUA_Beq = [1]

f_B = [ 0; 0; 0; 2000; 233;];
DMUB_Aeq = [5 50 50000 0 0];
DMUB_Beq = [1];

f_C = [ 0; 0; 0; 375; 220;];

DMUC_Aeq = [5 30 10000 0 0]
DMUC_Beq = [1]

f_D = [0; 0; 0; 1210; 245;];
DMUD_Aeq = [2 125 40000 0 0];
DMUD_Beq = [1];

f_E = [0; 0; ; 0; 230; 223;];

DMUE_Aeq = [4 20 15000 0 0];
DMUE_Beq = [1];

f_F = [0; 0; 0; 220; 235;];
DMUF_Aeq = [2 30 20000 0 0];
DMUF_Beq = [1]

f_G = [ 0; 0; 0; 190; 240;];
DMUG_Aeq = [1 18 4000 0 0];
DMUG_Beq = [1];

f_H = [ 0; 0; 0; 300; 230;];
DMUH_Aeq = [3 50 35000 0 0];
DMUH_Beq = [1]

f_I = [ 0; 0; 0; 205; 235;];
DMUI_Aeq = [2 30 20000 0 0];
DMUI_Beq = [1]

lb = zeros(5,1);
```

```
clc;
```

```
% Calculation of Technical
```

```
[X_A,FVAL_A,EXITFLAG,OUTPUT,lambdaA]=linprog(f_A,DMU,b,DMUA_Aeq,DMUB_Beq,lb);  
disp('A eficácia da DMU_A é:');  
disp(1/FVAL_A);
```

```
[X_B,FVAL_B,EXITFLAG,OUTPUT,lambdaB]=linprog(f_B,DMU,b,DMUB_Aeq,DMUB_Beq,lb);  
disp('A eficácia da DMU_B é:');  
disp(1/FVAL_B);
```

```
[X_C,FVAL_C,EXITFLAG,OUTPUT,lambdaC]=linprog(f_C,DMU,b,DMUC_Aeq,DMUC_Beq,lb);  
disp('A eficácia da DMU_C é:');  
disp(1/FVAL_C);
```

```
[X_D,FVAL_D,EXITFLAG,OUTPUT,lambdaD]=linprog(f_D,DMU,b,DMUD_Aeq,DMUD_Beq,lb);  
disp('A eficácia da DMU_D é:');  
disp(1/FVAL_D);
```

```
[X_E,FVAL_E,EXITFLAG,OUTPUT,lambdaE]=linprog(f_E,DMU,b,DMUE_Aeq,DMUE_Beq,lb);  
disp('A eficácia da DMU_E é:');  
disp(1/FVAL_E);
```

```
[X_F,FVAL_F,EXITFLAG,OUTPUT,lambdaF]=linprog(f_F,DMU,b,DMUF_Aeq,DMUF_Beq,lb);  
disp('A eficácia da DMU_F é:');  
disp(1/FVAL_F);
```

```
[X_G,FVAL_G,EXITFLAG,OUTPUT,lambdaG]=linprog(f_G,DMU,b,DMUG_Aeq,DMUG_Beq,lb);  
disp('A eficácia da DMU_G é:');  
disp(1/FVAL_G);
```

```
[X_H,FVAL_H,EXITFLAG,OUTPUT,lambdaH]=linprog(f_H,DMU,b,DMUH_Aeq,DMUH_Beq,lb);  
disp('A eficácia da DMU_H é:');  
disp(1/FVAL_H);
```

```
[X_I,FVAL_I,EXITFLAG,OUTPUT,lambdaI]=linprog(f_I,DMU,b,DMUI_Aeq,DMUI_Beq,lb);  
disp('A eficácia da DMU_I é:');  
disp(1/FVAL_I);
```



**UNIVERSIDADE ESTADUAL DO CEARÁ
FRANCISCO DALADIER MARQUES JÚNIOR**

**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA
AVALIAR O DESEMPENHO DE ISPs, ATRAVÉS DA
ANÁLISE ENVOLTÓRIA DE DADOS (DEA)**

**FORTALEZA – CEARÁ
2010**

FRANCISCO DALADIER MARQUES JÚNIOR

**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA AVALIAR O
DESEMPENHO DE ISPs, ATRAVÉS DA ANÁLISE ENVOLTÓRIA DE
DADOS (DEA)**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual do Ceará - UECE como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Jorge Luiz de Castro e Silva

Co-orientador: Prof. Dr. Antônio Clécio Fontelles Thomaz.

FORTALEZA – CEARÁ

2010

M357e

Marques Júnior, Francisco Daladier

Estimação de uma fronteira eficiente para avaliar o desempenho de ISPs, através da Análise Envoltória de Dados/ Francisco Daladier Marques Júnior. Fortaleza, 2010.

107 p.

Orientador: Prof. Dr. Jorge Luiz de Castro e Silva

Dissertação (Mestrado Acadêmico em Ciência da Computação) – Universidade Estadual do Ceará. Centro de Ciência e Tecnologia.

1. Análise Envoltória de Dados (DEA). 2. Programação Linear. 3. Avaliação de Desempenho. 4. Redes de Computadores. 5. ISPs. I.Universidade Estadual do Ceará. Centro de Ciência e Tecnologia.

CDD:001.6

FRANCISCO DALADIER MARQUES JÚNIOR

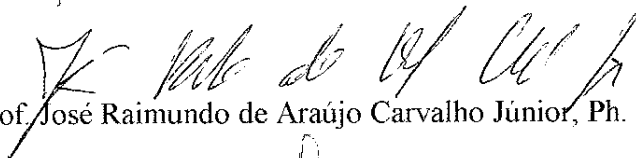
**ESTIMAÇÃO DE UMA FRONTEIRA EFICIENTE PARA AVALIAR O
DESEMPENHO DE ISPs, ATRAVÉS DA ANÁLISE ENVOLTÓRIA DE
DADOS (DEA)**

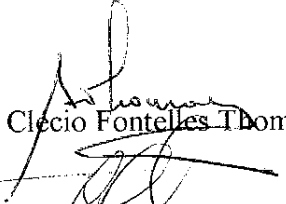
Dissertação de Mestrado submetida à comissão examinadora do Mestrado Acadêmico em Ciência da Computação, da Universidade Estadual do Ceará – UECE, como requisito parcial para obtenção de título de Mestre em Ciência da Computação. Aprovada pela comissão examinadora abaixo.

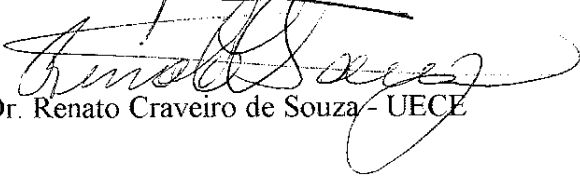
Aprovada em: 10/08/2010.

BANCA EXAMINADORA


Prof. Dr. Jorge Luiz de Castro e Silva (Orientador) – UECE


Prof. José Raimundo de Araújo Carvalho Júnior, Ph. D./ UFC


Prof. Dr. Antonio Clecio Fontelles Thomaz – UECE


Prof. Dr. Renato Craveiro de Souza - UECE

Agradecimentos

Em primeiro lugar agradeço a Deus pela força, discernimento e perseverança de lutar.

À minha esposa Maísa, pelas noites mal dormidas sem a minha presença, pelo compreensão, pelas palavras de incentivo, atenção, amor e carinho onipresentes;

Aos meus pais, Neide e Daladier, irmãos: D'Sávio e Raphaella e Mãe Corrinha que sempre me incentivarem a trilhar passos mais altos e me encorajaram a nunca desistir dos objetivos;

Aos meus avós, pelo carinho, apoio e orações. Maternos: vovó Santú (*in memorian*) e vovô Chico; Paternos: vovó Belinha e vovô Juarez (*in memorian*);

Aos primos e tios, pela preocupação com um futuro próspero em todos os pontos de vista;

Aos meus cães, Mayllow e Suzzy (*in memorian*), pelos momentos de alegria, amizade, dedicação, amor e companheirismo.

Ao meu orientador, Prof. Jorge Luiz, que depositou em mim a confiança antes esquecida. A ele agradeço a oportunidade de trilhar um caminho científico mais direcionado;

Ao meu co-orientador, Prof. Clécio Fontelles, pela dedicação e pelos ensinamentos sobre a DEA;

Ao IFPB, por minha liberação e investimento realizado;

Aos meus companheiros de apartamento que, na verdade, não foram apenas colegas, mas amigos de todas as horas a quem passo a citar: Alex e, em especial Walisson e Marcelo. Aos meus colegas Ênyo, pelo apoio e conselhos constantes; Marçal, pela alegria e descontração; Vigno; Fabiano, pelo comprometimento e seriedade; Gilzamir, além de outros;

Aos professores do MACC, em especial aos professores Marcial, Gustavo, Jerffeson, Valdísio, Celestino, André Ribeiro e André Moura, pelo comprometimento e lições em todos os sentidos;

Ao pessoal da secretaria e de apoio do MACC, em especial ao Marcos, Wágner, Elaine e o Auricélio (baixinho da limpeza);

Aos meus amigos e incentivadores no cotidiano: Fábio Gomes do IFPB, meu primo Osvaldo, meu amigo e veterinário Neto, meu colega de trabalho e amigo Ágio, Igor, Tiano, Samara, Fabiano (Fabinho) e Aninha;

Enfim, agradeço a todos que, de uma forma ou de outra torceram pelo meu sucesso nesta caminhada.

Resumo

A Análise Envoltória de Dados (DEA – *Data Envelopment Analysis*) é um método de estimação de fronteira Pareto-Koopmans eficiente, não-paramétrica, que auxilia tomadores de decisão a avaliarem Unidades de Tomada de Decisão (DMUs – *Data Making Units*) ou firmas sob uma ótica de estimação de pontos de referência, baseados na eficácia e produtividade. DEA tem sido largamente utilizada, nas últimas três décadas, na avaliação de escolas, setores de empresas, órgãos governamentais, empresas, organizações não-governamentais, entre outros para auxiliar a estas minimizarem a quantidade de insumos (entradas), a fim de produzir a mesma quantidade de produtos ou maximizar a quantidade de produtos (saídas), sem alterar a quantidade de insumos, através de um modelo de programação matemática ao solucionar um Problema de Programação Linear (PPL).

Ao analisar os ISPs (*Internet Services Providers* – Provedores de Serviço de Internet), nota-se que estes têm oferecido aos seus clientes, sub-serviços, principalmente pela falta de produtividade e eficácia nos processos empregados. Contudo, esta dissertação vislumbra contribuir, pontualmente, para que os ISPs possam oferecer melhores serviços aos seus clientes, bem como buscar reduzir vícios, com o intuito de obterem uma maior rentabilidade, ao ser aplicado a DEA, no julgamento de algumas variáveis, levantadas pontualmente, para solucionar tais problemas.

Palavras-Chave: Avaliação de desempenho. DEA. ISPs. Programação Linear. Redes de computadores.

Abstract

DEA (Data Envelopment Analysis) is a non-parametric estimation method of efficient Pareto-Koopman frontier which help the decision makers evaluate Data Making Units (DMUs) or firms under estimation from benchmarks based on efficiency and productivity. DEA have been broadly used in the past three decades in assessment from schools, enterprise sectors, governments, enterprises, NGO's (Non-Governmental Organization), and others to assist to minimize the inputs quantities to produce the same output quantities or to maximize the output quantities without changing the input quantities through of the mathematical programming to solve a Linear Programation Problem.

To analyze ISPs (Internet Service Providers) note that they offer their customers sub-services, due to lack of productivity and efficiency in the processes used. Hence, this dissertation sees occasionally contribute to the ISP's can offer better services to their customers, and seek to reduce defects in order to obtain a higher return when applied to the DEA in the trial of some variables raised promptly to resolve such problems.

Keywords: DEA. Computer Networks. ISPs. Linear Programming. Performance Evaluation.

Lista de Figuras

Figura 2.1. Espaço de busca de soluções do Simplex.	10
Figura 2.2. Envoltório de Dados	15
Figura 2.3. Medidas de eficiência e folgas de entrada [COELLI 1996].	15
Figura 2.4. Fronteiras CCR x BCC [COOPER et. al. 2007].	20
Figura 3.1. Mapa dos estados do Ceará e Paraíba com as cidades dos ISPs analisados.	26
Figura 3.2. Arquitetura proposta pelo trabalho.	29
Figura 3.3. Fluxograma algorítmico para estimação de uma fronteira eficiente DEA.	31
Figura 3.4. Fórmulas do modelo CCR/M/I aplicados a uma das DMUs analisadas.	32
Figura 3.5. Relatório da resolução de problemas de programação linear com o LINDO.	33
Figura 3.6. Fórmulas do modelo CCR/M/O aplicados a uma das DMUs analisadas.	34
Figura 3.7. Fórmulas do modelo CCR/E/I aplicados a uma das DMUs analisadas.	34
Figura 3.8. Fórmulas do modelo MBF aplicados a uma das DMUs analisadas.	37
Figura 3.9. Relatório das folgas e faltas do MBF relativo a uma das DMUs analisadas.	37
Figura 4.1. Saída do software proposto para o CCR/E/I.	41
Figura 4.2. Parceiros de Referência (<i>Benchmrks</i>) das DMUs.	41
Figura 4.3. Sumário de Eficiências Técnicas das DMUs calculadas no Win4DEAP.	42
Figura 4.4. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas.	42
Figura 4.5. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas.	42
Figura 4.6. Gráfico das Eficiências Técnicas do modelo CCR gerado pelo Solver DEA proposto.	43
Figura 4.7. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.	43
Figura 4.8. (a) Sumário das faltas das saídas e (b) Sumário das folgas das entradas, segundo o CCR/E/O.	44

Figura 4.9. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o CCR/E/O.	44
Figura 4.10. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/I.	45
Figura 4.11. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas, segundo BCC/E/I.	45
Figura 4.12.(a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/O.	46
Figura 4.13. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o BCC/E/O.	46
Figura 4.14. Gráfico das Eficiências Técnicas do modelo BCC gerado pelo <i>Solver</i> DEA proposto.	47
Figura 4.15. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.	48
Figura 4.16. Resultados do MBF calculados com o <i>Solver</i> DEA proposto.	48

Lista de Tabelas

Tabela 3.1. Entradas com os dados originais dos ISPs analisados.....	27
Tabela 3.2. Saídas com os dados originais dos ISPs analisados	27
Tabela 3.3. Matriz de correlação dos dados originais.	27
Tabela 3.4. Matriz de correlação definitiva dos dados das DMUs.....	31
Tabela 3.5. Dados das DMUs normalizados.....	32
Tabela 4.1. Ranqueamento das DMUs analisadas segundo CCR/E/I usando o DEA-Solver da Saitech.	43
Tabela 4.2. Eficiências técnicas, de escala e RVE's das DMUs analisadas.	47
Tabela 4.3. Ranqueamento das DMUs analisadas segundo BCC/E/I usando o DEA-Solver da Saitech.	47

Lista de Siglas e Abreviaturas

BCC	Barnes, Charnes e Cooper
BCC/E/I	BCC, modelo de envoltório e orientação à entrada
BCC/E/O	BCC, modelo de envoltório e orientação à saída
CCR	Charnes, Cooper e Rhodes
CCR/E/I	CCR, modelo de envoltório e orientação à entrada
CCR/E/O	CCR, modelo de envoltório e orientação à saída
CRS	Constant Return to Scale
DEA	Data Envelopment Analysis
DMU	Data Making Unit
ISP	Internet Service Provider
MBF	Modelo Baseado nas Folgas
MPSS	Most Productive Scale Size
PL	Programação Linear
PPL	Problema de Programação Linear
RNC	Retorno Não-Crescente
RND	Retorno Não-Decrescente
SCOOP	Scientific Computation of Optimal Programs
VRS	Variable Return to Scale

Sumário

Capítulo 1	1
Introdução	1
1.1. Definição do Problema.....	1
1.2. Objetivos.....	3
1.2.1. Objetivo Geral da Dissertação.....	3
1.2.2. Objetivos Específicos.....	3
1.3. Relevância.....	4
1.5. Organização da dissertação.....	6
1.6. Comentários	7
Capítulo 2	8
Fundamentação Teórica.....	8
2.1. Programação Linear.....	8
2.1.1. Histórico	8
2.1.2. Fundamentos	9
2.1.3. Algoritmo Simplex.....	10
2.2. Breve Histórico da DEA	11
2.3. Conceitos da DEA	13
2.3.1. Folgas.....	15
2.4. Modelos da DEA.....	16
2.4.1. CCR (Charnes, Cooper e Rhodes)	17
2.4.2. BCC (Banker, Charnes e Cooper).....	19
2.4.3. Modelo Aditivo de Folgas ou Modelo Baseado nas Folgas (MBF).....	22
2.4.4 Outros modelos da DEA	23

2.5 Comentários	24
Capítulo 3	25
3.1. Cenário.....	25
3.1.1. Dados Reais	25
3.2. Metodologia	27
3.2.1. Arquitetura proposta	29
3.3. Normalização dos Dados	31
3.3 Dados normalizados na DEA com seus respectivos cálculos	32
3.4 Comentários	38
Capítulo 4	39
Resultados e Discussões	39
4.1. Ambiente de Desenvolvimento	39
4.2. Resultados Obtidos	40
4.2.1. Resultados das DMUs analisadas segundo o CCR	40
4.2.2 Resultados das DMUs analisadas segundo o BCC	44
4.2.3 Resultados das DMUs analisadas segundo o MBF	48
4.3. Análise dos resultados obtidos	48
4.4. Comentários	49
Capítulo 5	50
5.1 Conclusões	50
5.2 Trabalhos futuros.....	51
Referências Bibliográficas	53

Capítulo 1

Introdução

Este trabalho apresenta uma dissertação, submetida para obtenção do grau de Mestre, no Programa de Pós-Graduação em Ciências da Computação, do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará.

Este capítulo está dividido desta forma: 1.1) apresenta o problema e a motivação do trabalho; 1.2) apresenta o objetivo geral e os objetivos específicos da dissertação; 1.3) mostra a relevância científica do trabalho; 1.4) apresenta como a dissertação está organizada, e; 1.5) mostra os comentários finais acerca do capítulo.

1.1. Definição do Problema

A Internet vem ganhando novos adeptos a cada dia estimulando o surgimento de novos serviços e de seus prestadores de serviços, denominados ISPs (*Internet Service Providers*) ou Provedores de Internet. Os ISPs estão interligados à Internet, através de outros Provedores de Serviço de Interconexão, chamados *backbones*, geralmente empresas de telecomunicações.

A Internet foi criada, comercialmente no início da década de 90, nos Estados Unidos, com a migração de uma rede, meramente, acadêmica e com propósitos de disseminação de conhecimentos. A grande transformação de interesses foi gerada com a criação da WWW (*World Wide Web*), pelo cientista britânico Tim Berners-Lee, radicado no CERN, em Genebra na Suíça, em 1989, que propôs a criação de páginas navegáveis via hipertexto para facilitar as limitações de velocidade da época. A invenção de Berners-Lee gerou um grande facilitador de publicações de ideias na Internet e, então, assim cientistas de todo o mundo puderam ficar a par do estado da arte de várias linhas de pesquisa.

No Brasil, a Internet chegou na conferência ECO-92, mas só começou a ser explorada comercialmente, na metade da década de 90, pois antes só existiam conexões acadêmicas. A primeira conexão, no Brasil, foi em 1988, no LNCC (Laboratório Nacional de Computação Científica), que trabalhava com uma conexão de 9.600 bps (bits por segundo) interligada com a BITNET.

A exploração comercial da Internet, no Brasil, começou na metade da década de 90, chegando ao Nordeste do Brasil, em 1996, com o provedor chamado Openline, situado em João Pessoa-Paraíba. A conexão do cliente, via ISP com a Internet, era através de modems que

evoluíram com o tempo, passando a trabalhar na capacidade máxima do canal telefônico, que é de, aproximadamente, 56,6 kbps (kilo bits por segundo), segundo o teorema de Shannon.

O rápido crescimento da Internet exigiu que os *sites* das empresas e de usuários se tornassem dinâmicos, i.e., não fossem mais meramente ilustrativos, mas que pudessem interagir com outras tecnologias, incorporando linguagens de programação e Bancos de Dados aos simples hipertextos, visando à venda de produtos na Web. Esta transformação criou o comércio eletrônico ou e-commerce.

O desenvolvimento de novos serviços e de *sites* cada vez mais especializados, impulsionou o desenvolvimento de novas tecnologias de conexão dos ISPs para os clientes, objetivando o aumento da velocidade nas conexões de última milha. Entre estas tecnologias podem ser citadas: ADSL (*Asymmetric Digital Subscriber Line*), Internet via cabo, Internet via satélite, Internet sem fio para pequenas (WLAN – Wireless LAN), médias (WMAN – Wireless MAN, compatíveis com o protocolo 802.11) e longas distâncias (WiMax, HSDPA (*High Speed Downlink Packet Access*), etc.), fibra óptica, entre outras.

Estas tecnologias fizeram com que os *backbones* investissem muito em sua infraestrutura para oferecer serviços mais rápidos, mais baratos e mais confiáveis para os ISPs. Os provedores de conexão tiveram, também, de investir em capacitação e melhoria na sua infraestrutura para atender um número cada vez maior de clientes conectados.

O preço de ligação de um ISP com um *backbone* varia muito de uma região para outra, principalmente pelo fato de que, em regiões metropolitanas, onde o poder aquisitivo é maior, tem-se uma concorrência maior entre os *backbones*, o que diminui o preço da conexão. Assim, em regiões mais remotas, esta concorrência se dá entre duas ou, no máximo, três empresas de telecomunicações, o que faz com que o preço da conexão, nestas regiões, torne-se muito alto. O interior das regiões Norte, Nordeste e Centro-Oeste do Brasil fazem parte desta realidade apresentando preços elevados.

A diminuição do preço do *hardware* e a interiorização dos provedores de Internet no país fizeram com que os negócios se expandissem, sendo que muitos destes provedores visam mais o lucro do que a qualidade do serviço oferecido. Em horário comercial, quando o tráfego de dados é intenso, os serviços oferecidos por muitos ISPs não chegam a atingir 20% ou 30% da capacidade contratada e, desta forma, os clientes são penalizados pela contratação de sub-serviços.

Mesmo assim, muitas ferramentas de divisão da largura de banda entre os clientes são utilizadas tanto em software, como em hardware; bem como o uso de proxies, para mascarar a velocidade de conexão do provedor para o cliente, entre outras. Contudo, aplicações P2P (*Peer to Peer*) muitas vezes burlam essas ferramentas, enquanto os usuários chamados de *heavy users* (usuários pesados) ficam com a maior parte da velocidade, fazendo com que os demais usuários fiquem com fatias de velocidade menores. Logo, fatores preponderantes, para que um ISP possa oferecer bons serviços aos clientes são a contratação de um bom profissional de administração de redes, um profissional para projetar e montar uma boa infra-estrutura física da rede, contratação de um *link* que atenda as exigências contratuais de seus clientes, uma boa rede de suporte técnico e um administrador de empresas competente que consiga trabalhar, com todas estas variáveis citadas, trazendo lucros para empresa.

Este trabalho visa a aumentar o nível de satisfação dos clientes e das empresas, empregando uma técnica de estimação de uma fronteira Pareto-Koopman [KOOPMANS 1951] de pontos de referências (*benchmarks*) eficientes, baseados na Análise Envoltória de Dados (DEA – *Data Envelopment Analysis*), proposta originalmente por Charnes, Cooper e Rhodes em 1978 [COOPER e ZHU 2004]. Entretanto, esta mesma metodologia foi usada para estimar a eficiência dos ISPs, localizados no interior do estado do Ceará e Paraíba, tanto no ponto de vista da satisfação do cliente, quanto da maximização dos lucros das empresas.

1.2. Objetivos

1.2.1. Objetivo Geral da Dissertação

Avaliar as eficiências dos provedores de Internet com DEA, através de ferramentas computacionais desenvolvidas para este propósito, com o intuito de prover fronteiras eficientes Pareto-Koopman sob várias óticas, i.e., aumentar o nível de satisfação dos clientes e das empresas usadas para estimação das fronteiras.

1.2.2. Objetivos Específicos

- a) Estudar a DEA, seus modelos, formas, orientações e aplicabilidades;
- b) Investigar e identificar as principais ferramentas computacionais especializadas em DEA quer sejam livres ou proprietárias;
- c) Levantar os dados relevantes com os diversos ISPs que farão parte do modelo;
- d) Especificar e desenvolver um software para computar as fronteiras de eficiência, segundo o modelo, forma e orientação que se deseja;

- e) Apresentar os resultados, obtidos através da DEA e do software especialista desenvolvido, a respeito das empresas em questão;
- f) Mostrar a importância da abordagem DEA do ponto de vista econômico, além do ponto de vista tecnológico e especialista.

1.3. Relevância

Por ser a Internet uma realidade no cotidiano de pessoas e empresas ao redor do globo, muitos destes entes nortearam-na com uma responsabilidade descomunal, devido ao ganho de produtividade oferecido por ela. Assim, é de extrema importância prover serviços satisfatórios aos usuários que a utilizam na ponta; por conseguinte este serviço, em muitas ocasiões, é ofertado de maneira insatisfatória pelos prestadores de serviços denominados ISPs que, em grande parte, à medida que novos clientes aderem à utilização do serviço, tanto antigos, quanto novos clientes se depararão com uma situação cada vez mais corriqueira que é a diminuição na vazão ou velocidade contratada em detrimento de um lucro maior.

Por sua vez, a simples melhora no serviço, oferecido aos clientes pode desencadear um rápido processo falimentar dos ISPs, visto que estes têm de desembolsar uma quantidade fixa, mensalmente, para o pagamento do *link* à empresa de interconexão. Contudo, com a aplicação da Análise Envoltória de Dados (DEA), pode-se estimar uma fronteira de eficiência que pode contemplar as duas óticas com presteza, ou seja, tanto pode agradar o cliente com a maximização do desempenho por ele contratado, quanto pode buscar uma melhora relativa aos lucros dos ISPs.

Portanto, a solução do problema proposto é relevante, pois as firmas necessitam de melhor planejar seus processos produtivos, com a finalidade de perdurarem no mercado; além de, também, poderem projetar a entrada de novos clientes, de maneira que clientes antigos e novos saíam satisfeitos com os serviços por elas oferecidos.

1.4. Trabalhos relacionados

A Análise Envoltória de Dados (DEA) tem sido largamente utilizada, mais especificamente na dissertação de PhD. de G. Tavares, onde desenvolveu um apanhado bibliográfico da DEA, de 1978 a 2001 quando, até aquela data, existiam 3600 artigos escritos, além de diversos livros. Assim, mais de 1600 autores, em cerca de 42 países incluíam a avaliação de forças policiais, as junções econômicas e performances ambientais de empresas públicas na Europa, bem como o progresso social das sociedades árabes do Oriente Médio e Norte da África [FERREIRA e GOMES 2009].

Outra rica fonte a respeito da DEA é o sítio de Ali Emrouznejad, intitulado de DEAZONE. Tal sítio contém tutoriais, uma lista de 320 dissertações, várias publicações e livros a respeito da DEA.

Conforme citado, a DEA teve sua concepção no trabalho de dissertação de PhD. de Ewardo Lao Rhodes em 1978 [CHARNES, COOPER e RHODES 1978]. Em seu estudo, Rhodes avaliou o desempenho de um programa educacional denominado “*Follow Through*” (em português: Siga totalmente) e, neste mesmo trabalho, foram criados os termos DMUs (*Data Making Units* – Unidades de Tomada de Decisão), além das entradas ou insumos (*inputs*) e produtos ou saídas (*outputs*), sem esquecer, também, de toda uma base formal.

No Brasil a DEA vem recebendo, também, uma atenção crescente, destarte o trabalho de Villarroel et. al. (1999) que mostrou uma proposta para modelar indicadores de saúde das Microrregiões do Estado de Santa Catarina, usando DEA. O trabalho de MACEDO et. al. (2005) avaliou a eficiência dos investimentos em TI (Tecnologia da Informação) de empresas de tecnologia usando a DEA; Resende Neto (2006) trouxe uma análise de performance de fundos de investimento no país através da DEA.

Quintanilha e Ho (2006) publicaram um estudo sobre DEA para comparar a eficiência das ações de monitoramento de risco de incêndio, em municípios da Região Amazônica brasileira.

Chaves e Thomaz (2008) avaliaram o desempenho das agências da Previdência Social na região metropolitana de Fortaleza, Ceará, além de outras agências do interior do estado do Ceará, avaliando um número considerável de 22 DMUs. Tal estudo teve papel preponderante para melhoria no atendimento, destas agências, ao ser utilizada a DEA para gerar uma fronteira eficiente entre as agências analisadas, além de uma projeção de como as agências ineficientes poderiam acompanhar os seus parceiros de referência.

Existe uma gama de softwares que consegue resolver os cálculos das DMUs, através da DEA, usando a programação matemática (linear), dos quais podem ser citados: o MATLAB, que trabalha com programação matemática, centrado em operações, envolvendo vetores e matrizes; o LINDO, que resolve problemas de programação linear através do algoritmo Simplex; o LINGO, que resolve problemas, usando a programação linear, não-linear e inteira. Dentre estes podemos, ainda, citar o projeto nacional, desenvolvido por professores e alunos da Universidade Federal de Viçosa-MG, denominado PROLIN, que resolve sistemas através da programação linear na Web.

Por outro lado existe, também, uma série de softwares, especialistas na DEA, sendo que vários destes livres, podem ser citados: o DEAP, os nacionais SIAD, IDEAL que é o único dos softwares a mostrar uma fronteira tridimensional e o SAED. Explorando a DEA, comercialmente, existem vários softwares, tais como: o DEA Excel Solver, o PIMSoft, o Frontier Analyst, dentre outros. Um dos principais softwares gratuitos pode ser citado: o DEA-Excel que consegue trabalhar com até duzentas DMUs e vinte entradas e saídas.

Várias ferramentas foram propostas, a fim de se checar a vazão fim a fim entre hosts que compõem uma rede, em especial a Internet. Estas ferramentas foram desenvolvidas baseadas em técnicas de medições passiva e ativa [BARFORD e SOMMERS 2004]. Assim, a monitoração de tráfego é dada, através do registro passivo de pacotes em um enlace, enquanto as medições ativas de desempenho usam o envio de pacotes, denominadas de sondas, de medição [ZIVIANI e DUARTE 2005].

Entretanto, estas técnicas levam em conta, apenas, o lado do desempenho da rede de computadores, mas não avaliam o desempenho econômico. Por conseguinte, os clientes dos ISPs podem recorrer a ferramentas Web para conseguirem analisar a velocidade com a qual estão conectados, em um dado momento e, com a finalidade de saber se os serviços, por eles contratado, estão em conformidade. Entre estas ferramentas Web podem ser citadas: o velocímetro do ISP RJNET, o velocímetro da Beltrônica, etc.

O clássico trabalho de Jain (1991) relaciona as três principais técnicas, aplicadas à análise de desempenho em redes de computadores. São elas: a simulação, a modelagem analítica e a experimentação.

1.5. Organização da dissertação

Esta dissertação está organizada em 5 (cinco) Capítulos e suas respectivas referências bibliográficas.

O primeiro Capítulo é uma introdução à dissertação, em que mostra uma contextualização do problema, a motivação, os objetivos geral e específicos a serem alcançados, além da relevância do problema e as suas contribuições. Tal capítulo remete-se a uma revisão bibliográfica do assunto abordado, apresentando a DEA, sendo usado para estimação de fronteiras eficientes em várias áreas distintas, além de mostrar a estrutura da dissertação.

O segundo Capítulo apresenta os conceitos, as técnicas empregadas e as fundamentações teóricas usadas neste trabalho. Neste capítulo é apresentada a programação

linear, com seu histórico e fundamentos, além de abordar o algoritmo Simplex. Também é formalizado o problema de satisfação dos clientes, versus a maximização dos lucros, pelas DMUs (*Data Making Units*) ou Unidades de Tomada de Decisão em estudo. No caso, os ISPs. É apresentada, também, a essência da DEA desde a sua concepção. Neste, são conhecidos os principais modelos da DEA, com suas respectivas formas e orientações, bem como mostra outros modelos DEA.

O terceiro capítulo é devotado ao cenário, utilizado na pesquisa e na metodologia usada para o desenvolvimento deste trabalho, acompanhado pela arquitetura proposta, do fluxograma algorítmico utilizado e mostrando o processo de normalização dos dados, passo este importante para se alcançar uma maior veracidade nos cálculos DEA.

No quarto capítulo serão mostrados os resultados, obtidos com o uso do *Solver* DEA proposto, além de outros softwares auxiliares, para se obterem resultados mais apurados e, também, para servirem como provas dos resultados alcançados pelo *Solver* DEA proposto.

Finalmente, o quinto capítulo encerra a dissertação com as conclusões do trabalho realizado e as seguiu das respectivas futuras pesquisas.

1.6. Comentários

Este capítulo mostrou o problema que se quer resolver, os objetivos que devem ser alcançados ao término do trabalho dissertativo, a relevância do problema para os ISPs e outras empresas em geral, além de expor a estrutura da dissertação.

Ainda foi apresentada a Análise Envoltória de Dados (DEA) que tem se tornado pertinente, ao analisar o desempenho quantitativo de empresas ou firmas ou Unidades Tomadores de Decisão (DMUs), acerca das últimas três décadas. Além disso, o problema de análise de desempenho de redes de computadores é bastante conhecido na literatura e empregado para prover maior eficácia no transporte fim a fim de pacotes inter-redes. Assim, este capítulo foi remetido às principais pesquisas e ferramentas, usadas nestas áreas, nos últimos anos.

Capítulo 2

Fundamentação Teórica

Este capítulo abordará os aspectos teóricos que fundamentam a proposta defendida neste trabalho. Na primeira parte do capítulo, serão apresentados os conceitos e fundamentos da programação linear, passando pelo algoritmo Simplex. Na segunda parte, será mostrado um histórico da DEA. A terceira parte tratará de alguns conceitos usados pela DEA. A quarta parte apresentará os formalismos da DEA, direcionada, principalmente, para os modelos DEA, usados neste trabalho; bem como serão apresentados alguns modelos adicionais.

2.1. Programação Linear

Nesta seção será mostrado um apanhado da programação linear e do algoritmo Simplex, usado para solucionar PPL's.

2.1.1. Histórico

A programação linear é baseada na álgebra linear. Assim são usados cálculos, iterativamente, para solucionar problemas através de algoritmos computacionais. Problemas de Programação Linear (PPL) são problemas de otimização, sendo que a função objetivo e as suas restrições são lineares.

De acordo com Lins e Calôba (2006), dois trabalhos científicos influenciaram o desenvolvimento da Programação Linear (PL). Foram eles:

1. Desenvolvimento da Teoria dos Jogos por John Von Neumann em 1928;
2. Criação da Análise insumo-produto por Leontief em 1936.

Outros trabalhos científicos surgiram neste período, sendo formulados sob a forma de restrições, aumentando o interesse da comunidade científica para solucionar estes problemas. Daí, surgiram os PPL, tais como: o problema da organização e planejamento de produção, o problema do transporte, o problema da dieta, entre outros.

A proposição e a consequente resolução destes problemas impulsionou a consolidação de uma nova área de pesquisa, denominada de Pesquisa Operacional, que usa a PL para solucionar problemas de otimização.

As luzes da pesquisa operacional começaram a clarear o mundo científico na 2ª Guerra Mundial nos EUA. Nesta época, foi criado um projeto chamado de SCOOP (*Scientific*

Computation of Optimal Programs) que avaliava a viabilidade da aplicação de métodos matemáticos em problemas orçamentários e de planejamento militar. Um dos membros do SCOOP era George Dantzig, que no mesmo ano que se iniciou o projeto, 1947, juntamente com sua equipe de desenvolvimento, desenvolveram um método de solução denominado Simplex.

Com o passar do tempo, o aumento do interesse pela PL cresceu entre os cientistas, bem como pelas áreas dos PPL. Assim, podem ser encontradas soluções com PL para problemas de meio ambiente, transporte, energia, problemas sociais, industriais e em outras áreas.

Outro marco importante na PL foi o desenvolvimento do algoritmo dos pontos interiores por Karmakar (1984). O algoritmo dos pontos interiores era um algoritmo polinomial, para resolução de PPL's, que obtém resultados com eficiência e eficácia. O objetivo do algoritmo dos pontos interiores é transformar as restrições de desigualdade em restrições de igualdade com a incorporação de variáveis de folga e do uso de uma barreira logarítmica à função objetivo, para evitar a negatividade das variáveis de folga, fazendo com que a solução esteja na região de soluções viáveis.

Atualmente, a PL vem auxiliando outras ciências em busca de se otimizar os resultados através da solução de vários PPL's. Uma das técnicas que usam a PL é a DEA, que se utiliza do Simplex para estimar sua fronteira de eficiência.

2.1.2. Fundamentos

Segundo Lins e Calôba (2006), o uso de modelos de PL compreende as etapas de formulação do PPL, resolução e validação dos resultados. A primeira parte é a representação matemática do problema proposto e, nesta etapa, serão definidas as variáveis e suas relações, reconhecidas as restrições relevantes e expresso(s) o(s) objetivo(s).

Assim, para a resolução de PPL's, são necessários os seguintes elementos:

- Variáveis de decisão: são relevantes ao problema, passíveis de quantificação e disponíveis;
- Função Objetivo: é uma função, produto de coeficientes pelas variáveis de decisão, ao qual o problema deseja otimizar, ora através da maximização, ou da minimização;
- Restrições: são os elementos restritivos que todo problema possui, tal como restrições de capital para investimento ou de escassez de recursos.

2.1.3. Algoritmo Simplex

O Algoritmo Simplex foi proposto por George Dantzig (1963), em 1947, sendo desenvolvido pela equipe do SCOOP. O nome do algoritmo veio do conceito de um simplex, i.e., uma figura geométrica de $N+1$ vértices em N dimensões, i.e, um segmento de reta sobre uma reta, ou um triângulo sobre um plano.

O algoritmo Simplex busca as chamadas soluções básicas viáveis, que são aquelas que se situam nos vértices da região limitadora de soluções ótimas. Assim, se o PPL tem solução ótima, haverá uma solução básica viável que será uma solução ótima [LINS e CALÔBA 2006].

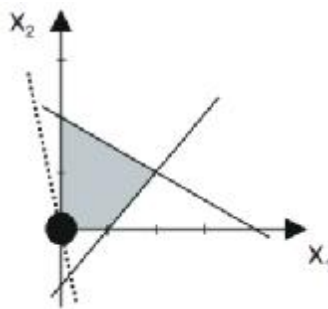


Figura 2.1. Espaço de busca de soluções do Simplex.

A sequência de passos lógicos finitos, usados pelo Simplex, são [LINS e CALÔBA 2006]:

- Buscar a primeira solução básica viável;
- Checar se existem vértices adjacentes viáveis com possibilidade de melhoria da função objetivo, isto é, executar um teste de otimalidade;
- Movimentar a solução para um vértice melhor, da seguinte maneira:
 - o Examinar qual dos $m-1$ vértices adjacentes otimiza melhor a função objetivo, sendo que m é o número de restrições do problema;
 - o Deter o acréscimo da variável que entrou na base do limite da restrição (i.e, o vértice);
 - o Solucionar o problema para este vértice com o pivoteamento das equações;
- Aplicar o teste de otimalidade;
 - o Se a solução for ótima, parar a execução do algoritmo;

- Se a solução não for ótima, realizar uma nova iteração;
- Iterar o algoritmo até encerrar o problema.

O algoritmo Simplex se dá bem na maioria dos problemas propostos, entretanto um PPL pode admitir vários tipos de soluções, a saber:

- Uma única solução ótima;
- Infinitas soluções ótimas;
- Problema inviável;
- Problema ilimitado.

2.2. Breve Histórico da DEA

A DEA se confunde com a Teoria da Produção, pois ambas buscam a eficiência das empresas que são submetidas às análises. A DEA surgiu de conceitos da produção microeconômica que, segundo Ferreira e Gomes (2009), foram registrados no livro de Johann-Heinrich Von Thünen (1783-1850), intitulado de “O Estado Isolado” e, tal obra, tratava de melhoria na produção agrícola.

Jon Von Neumann, notável cientista com contribuições em muitas áreas, em seu trabalho [NEUMMAN 1945], apresentou um modelo geral de equilíbrio econômico, mostrando que os bens são produzidos por si, numa espécie de processo circular, e que o número de processos técnicos de produção pode ser maior do que o de mercadorias. Este trabalho usou a ideia de retornos constantes. Ele dizia, também, que as quantidades usadas e as produzidas dos bens no qual os processos de produção dependem devem ser, obviamente, não negativas.

Lins e Calôba (2006) apontam que, em 1951, Debreu introduziu uma medida radial de eficiência técnica, denominado de coeficiente de utilização de recursos. Tal medida radial pretende alcançar a máxima redução equiproporcional de todas as entradas ou a máxima expansão equiproporcional de todas as saídas; sendo tanto as entradas, como as saídas, independentes de unidades. O grande problema deste modelo é que o mesmo pode ser ineficiente na definição de Koopman-Pareto.

Outro trabalho seminal foi o de Dantzig (1963) que influenciou um ramo da ciência em especial, denominado Pesquisa Operacional, muito importante para obter otimização dos processos produtivos. Dantzig mostrou os fundamentos do método Simplex, que através de iterações no espaço de soluções; em especial, nos vértices, busca uma solução possível se ela

existir. O Simplex trabalha com a introdução de variáveis artificiais, resultando num problema auxiliar na forma canônica, daí o Simplex é empregado. Este método consiste em uma sequência de operações pivôs, referenciadas como Fase I que resultarão em uma sucessão de formas canônicas, a fim de encontrar uma solução possível num espaço de soluções. Caso a forma canônica produza cada solução, então o Simplex é novamente aplicado em uma segunda sucessão de operações de pivôs chamada de Fase II. Caso a solução não encontre uma solução o algoritmo terminará em um número finito de iterações.

Outro conceito empregado é da fronteira de eficiência, baseada no trabalho de Koopmans (1951), que foi fortemente influenciado pelo trabalho do economista, sociólogo e engenheiro ítalo-suíço Vilfredo Pareto. Por isso a menção a fronteira de otimalidade de Pareto-Koopmans, também conhecida como fronteira ótima de Pareto, é baseada no pensamento abaixo:

“Situação hipotética em que ninguém pode alterar sua situação social, buscando uma posição mais cômoda, sem que isso provoque uma mudança de outra pessoa para uma posição que não lhe agrade”.

Como, novamente, exposto por Ferreira e Gomes (2009), mostrou-se, neste trabalho, a grande importância do cientista James Farrel em seu trabalho: “The Measurement of Productive Efficiency” em 1957, onde Farrel desenvolveu métodos para avaliar a produtividade, através dos conceitos de análise de atividades; mas tal trabalho limitou-se, usando apenas um único produto, pois, ao se trabalhar com múltiplos produtos, o modelo tornava-se ineficiente para a aplicação de um grande conjunto de dados. Tal trabalho lhe valeu o prêmio Nobel de economia.

A DEA apareceu como método de estimação de fronteira eficiente, a partir da década de 70, no esforço de Edwardo Lao Rhodes em obter o grau de PhD [COOPER et. al. 2007]. Deste modo, em 1978, o termo DEA apareceu em um relatório chamado de: “*A Data Envelopment Analysis Approach to Evaluation of the Program Follow Through Experiment in U.S. Public School Education*”. Tal programa avaliava a eficiência de escolas públicas que pertenciam ao programa educacional *Follow Through* para alunos carentes, tornando-se um marco, para aplicar princípios de projetos estatísticos, a fim de mensurar um conjunto de escolas que faziam parte deste projeto. Tal trabalho introduziu o termo DMU (*Data Making Unit*), onde, naquele caso, eram as escolas avaliadas. Por ter utilizado um grande conjunto de dados, o trabalho de Rhodes não teve nenhum problema com as variáveis de entrada e saídas usadas. Assim, a DEA corrigiu os erros oriundos do trabalho de Farrel.

2.3. Conceitos da DEA

Por ser um modelo de estimação de fronteira de produção eficiente, a DEA baseia-se na premissa básica da produção, onde um conjunto de insumos unidos serão transformados através de processos produtivos, com o intuito de gerar produtos (saídas), para obter o maior lucro possível. Sabe-se que, nem sempre, esta produção trará lucros para as empresas que os manufaturam, devido a diversos fatores.

As empresas, ao se utilizarem dos processos produtivos empregados, podem ser eficientes se atingirem os objetivos propostos, sem levar em conta os recursos utilizados; ou seja, tais empresas podem ser eficientes, mas não lucrativas. Outro conceito é de produtividade, que está diretamente ligado ao modo de como os recursos de uma empresa estão sendo empregados, com a finalidade de obter um produto. Este conceito de produtividade se dá pela relação entre a produção com o insumo usado:

$$\frac{\text{Produção}}{\text{Insumo}} \quad (1)$$

A DEA, diferentemente do método de Farrel, não se limita a um único insumo e a um único produto, usando, assim, um número qualquer de entradas (insumos) e saídas (produtos) com o intuito de avaliar as empresas do ponto de vista da eficiência no emprego das entradas e saídas. O objetivo da DEA é fazer com que as DMUs possam minimizar a quantidade de entradas, para se obter a maximização das saídas, objetivando um maior lucro para as firmas avaliadas, além de definir quais das empresas avaliadas estão trabalhando com maior produtividade, pois, desta forma, a DEA poderá calcular como as empresas mal avaliadas poderiam trabalhar como as outras que são suas parceiras de referência (*benchmarks*).

Outro conceito de suma importância é o de eficiência técnica que compara o que foi produzido por uma unidade com o que poderia ser produzido [FERREIRA e GOMES 2009], da seguinte forma:

$$\frac{\text{Produção}}{\text{Insumo}} \text{ realizada, comparada com } \frac{\text{Produção}}{\text{Insumo}} \text{ a mais adequada}$$

Em relação aos dados, utilizados pela DEA, na entrada ou na saída, não se têm nenhuma restrição referentes às medidas adotados, podendo estes ser uma mescla entre preços, vazão de conexões fim a fim, funcionário num dado setor, quantidade de peso em quilogramas, entre outros.

Por outro lado as DMUs não necessitam ser, exclusivamente, empresas ou corporações, mas podem ser setores de empresas e/ou órgãos governamentais, filiais, escolas, órgãos governamentais, etc. Assim, a DEA pode ser usada, sob qualquer ótica, para avaliar um conjunto de entes produtivos, através de suas variáveis de entrada e de saída.

Pode-se afirmar que a DEA é um método matemático não paramétrico, pois não se utiliza de inferências estatísticas, nem se apega a medidas de tendência central, testes de coeficientes ou formalizações de análise de regressões [FERREIRA e GOMES 2009]. A ideia central da DEA, segundo COOPER et. al. (2007) é a avaliação de performance de atividades ou organizações, através da análise de eficiência de unidades, centrados em conceitos como produtividade e eficiência técnica. DEA usa técnicas como a programação matemática, que trata um grande número de variáveis e restrições. Isso relaxa os requisitos que são, frequentemente, encontrados, quando se está limitado à escolha de poucas entradas e saídas, porque as técnicas empregadas encontram dificuldades.

Depois que uma fronteira linear de eficiência, dividida por partes (*linear piecewise frontier*), com características convexas, for gerada, serão visualizadas DMUs que estão nesta fronteira e outras que estão abaixo da fronteira e que têm de alcançar seus parceiros de referência (*benchmarks*); ou através da minimização das entradas, sem a diminuição das saídas, ou a maximização dos lucros (saídas), sem aumentar as entradas, caracterizando o chamado movimento radial.

Na figura 2.2 é apresentado o modelo de envoltório de dados, baseado nas saídas produzidas por cada empresa. Deste modo, as posições representadas pelas empresas 1, 2 e 4 mostram que estas estão formando uma fronteira de eficiência e, assim, a empresa 3 está abaixo desta fronteira, precisando aumentar alguma(s) medida(s) para alcançar o nível de performance das empresas concorrentes. A figura a seguir ainda mostra que a empresa 2 é o parceiro de referência (*benchmark*) da empresa 3.

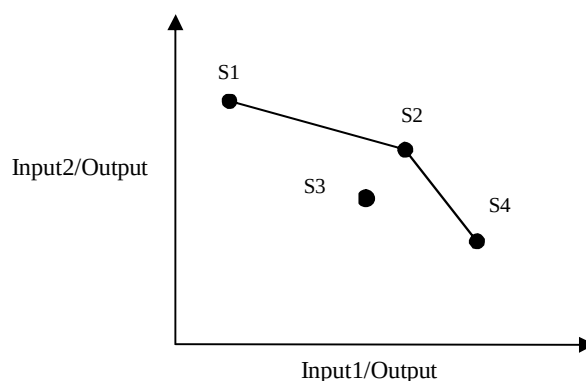


Figura 2.2. Envoltório de Dados

2.3.1. Folgas

A forma da fronteira não-paramétrica linear, na DEA, pode causar dificuldades na medida de eficiência [COELLI 1996]. Este problema é gerado, devido às seções (pedaços ou vizinhos ou *peers*), usadas na fronteira linear paralelas aos eixos, o que não ocorrem em muitas funções paramétricas.

No método de Farrel, as medidas de eficiência técnica entre duas DMUs A e B são, respectivamente, OA'/OA e OB'/OB . Mas, como se pode observar, na figura a seguir, deve-se questionar se o ponto A' é eficiente, desde que se poderia diminuir mais a quantidade da entrada x_2 (usada pela quantidade CA'), para produzir a mesma saída. Este conceito é chamado de folga que, tanto pode ser de entrada, sendo chamada de excesso; como de saída, que é chamada de insuficiência de saída ou falta.

Na figura 2.3, a folga de entrada (excesso), associada ao ponto A' , é CA' da entrada x_2 . Quando existem mais do que uma entrada e/ou saídas, o cálculo das folgas não é uma tarefa simples. Adiante, serão mostrados como podem ser calculadas estas folgas nas entradas e nas saídas.

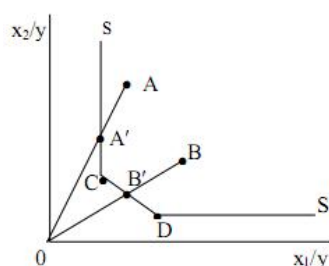


Figura 2.3. Medidas de eficiência e folgas de entrada [COELLI 1996].

2.4. Modelos da DEA

DEA, por ser um método eficiente para mensurar a eficiência das corporações, pressupõe tal fronteira sob duas óticas: i) diminuindo a quantidade de insumos, para produzir o mesmo produto, e; ii) aumentando a quantidade de produtos, mantendo a mesma proporção de insumos. Todas essas suposições são baseadas na ideia da Fronteira de Pareto.

A notação da DEA dispõe que os dados estão dispostos em K entradas e M saídas em cada uma das N firmas ou DMUs. A matrix KxN é chamada de matriz de entradas, sendo representada por X, e a matriz MxN é chamada de matriz de saídas que, sendo representada por Y, formam os dados de todas as N DMUs [COELLI 1996].

Uma das formas iniciais da DEA era através da forma fracionária, assim, para cada DMU deve-se obter uma medida, através de todas as saídas e sobre todas as entradas, tal como:

$$u' y_i / v' x_i \quad (2)$$

Onde u é um vetor Mx1 de saídas ponderadas (ou pesos), e v é um vetor Kx1 de entradas ponderadas. Tal suposição resultou em um modelo de programação matemática fracionária, para selecionar pesos ótimos de u e de v das DMUs, que pode ser solucionado através das fórmulas a seguir:

$$\max_{u,v} (u' y_i / v' x_i) \quad (3)$$

s.a.

$$u' y_j / v' x_j \leq 1, j = 1, 2, \dots, N \quad (4)$$

$$u, v \geq 0 \quad (5)$$

O problema da programação matemática fracionária é que ele tem um número infinito de soluções. Tal dificuldade foi contornada, através do uso da programação linear, mais especificamente pelo método Simplex, que resultou na formulação original da DEA, denominada de modelo CCR, uma alusão às iniciais dos mentores Charnes, Cooper e Rhodes.

Um outro problema encontrado da programação matemática fracionária, é que ela não garante que a restrição de não negatividade (5) gerará resultados positivos.

Deve ser lembrado que a razão entre as saídas virtuais pelas entradas virtuais não pode exceder o valor da unidade, ou seja, os valores têm de ser menores ou iguais a 1 (≤ 1).

Nas próximas seções serão mostradas as principais orientações da DEA, que são os modelos CCR e BCC (Barnes, Charnes e Cooper), além do modelo aditivo de folgas e, finalmente, serão mostrados alguns modelos complementares.

2.4.1. CCR (Charnes, Cooper e Rhodes)

O modelo CCR foi o modelo proposto, originalmente em 1978, pelos cientistas Charnes, Cooper e Rhodes, por isso esta orientação tem as iniciais destes. Este modelo pressupõe Retornos Constantes de Escala (RCE) ou CRS (*Constant Return to Scale*), i.e., que as entradas e as saídas são proporcionais entre si.

Enzo Mariano et. al. (2006) evidenciaram que o CCR desconsidera ganhos e escala quando calcula a eficiência. Portanto, a eficiência relativa de uma DMU é obtida através da divisão entre a sua produtividade e maior produtividade dentre as DMUs da observação. Assim, o formato da fronteira do modelo de eficiência CCR é uma reta que forma um ângulo de 45°.

Pelos problemas que foram mostrados da programação matemática fracionária, foi proposto o modelo de programação linear primal, através do método Simplex, para resolver problemas de otimização, obtido com a combinação linear das n variáveis de entrada e saída.

Na forma primal existem m restrições e estas restrições encontram-se em um limite superior (*upper bound*), em uma combinação linear de n variáveis. Na forma dual, estas mesmas restrições encontram-se em um limite inferior (*lower bound*). Todavia, ambas as formas geram, como solução, um vetor de s valores que alcançam os valores da função objetivo.

Assim, foi proposto o modelo CCR que transformou o problema de programação matemática fracionária num problema de programação matemática linear. Contudo, tal modelo possui tanto a forma primal, que também é chamada de modelo dos multiplicadores, como a forma dual, chamada de modelo de envoltório ou envoltório. Por isso, o nome Análise Envoltória de Dados.

Outro ponto a ser tocado é que todos os modelos, na DEA, possuem orientações para entrada e para saída, sendo que, em ambas as orientações, bem como nas formas dos multiplicadores e de envoltórios, são gerados valores idênticos. A única diferença considerável se dá em termos computacionais, pois o modelo de envoltório possui menos restrições do que o modelo dos multiplicadores, i.e., no modelo dos multiplicadores existem $N+1$ restrições, enquanto que no modelo de envoltório existem, $K+M$ restrições.

Abaixo, serão mostradas as fórmulas da DEA para o CCR, tanto para o modelo dos multiplicadores, quanto para o modelo de envoltório, seguindo ambas as direções (entradas e saídas). São elas:

Modelo dos Multiplicadores – Primal

Orientação a Insumo:

$$\text{Maximizar Eficiência} = \sum_{j=1}^m m_j y_{jo} \quad (6)$$

Sujeito a:

$$\sum_{i=1}^r v_i x_{io} = 1 \quad (7)$$

$$\sum_{j=1}^m m_j y_{jk} - \sum_{i=1}^r v_i x_{ik} \leq 0 \quad (8)$$

$$m_j, v_i \geq 0 \quad (9)$$

Onde v e μ são matrizes de entradas e saídas respectivamente. Vale ser ressaltado que tanto os modelos Primais, quanto Duais, em ambas as orientações, produzirão o mesmo resultado no modelo CCR.

Orientação a Produto:

$$\text{Minimizar Eficiência} = \sum_{i=1}^r v_i x_{io} \quad (10)$$

Sujeito a:

$$\sum_{j=1}^s m_j y_{jk} = 1 \quad (11)$$

$$\sum_{j=1}^s m_j y_{jk} - \sum_{i=1}^r v_i x_{ik} \leq 0 \quad (12)$$

$$m_j, v_i \geq 0$$

Modelo dos Envoltórios – Dual

Orientação a Insumo:

$$\text{Minimizar Eficiência} = q \quad (13)$$

Sujeito a:

$$q x_{io} - \sum_{k=1}^n l_k x_{ik} \geq 0 \quad (14)$$

$$\sum_{k=1}^n l_k y_{mk} - y_{mo} \geq 0 \quad (15)$$

$$l_k \geq 0 \quad (16)$$

Orientação a Produto:

$$\text{Maximizar Eficiência} = f \quad (17)$$

Sujeito a:

$$x_{io} - \sum_{k=1}^n x_{ik} l_k \geq 0 \quad (18)$$

$$\sum_{k=1}^n y_{mk} l_k - f y_{mo} \geq 0 \quad (19)$$

$$l_k \geq 0$$

Onde a eficiência θ será obtida da razão $1/f$. Nota-se, também, que todas as restrições não excedem a unidade, nem são menores que zero, caracterizando, assim, as porcentagens referentes às eficiências técnicas de cada uma das DMUs em questão.

Outra informação que poderá ser avistada nos vários livros, artigos e outros da DEA é que, geralmente, é utilizado o formato Modelo/Forma/Orientação para simplificar a descrição das fórmulas, i.e., ao usar o CCR na forma dos multiplicadores e, com a orientação para os produtos ou saída, a descrição das fórmulas seria CCR/M/O, onde o I é relativo às entradas ou insumos e o O é relativo às saídas ou produtos. No caso, quando se usa o modelo BCC, troca-se o CCR por BCC. Se for usada a forma de envoltório, é trocada a letra M pela letra E; entretanto, se for usada a orientação de entrada em detrimento da de saída, usa-se o I em vez do O.

2.4.2. BCC (Banker, Charnes e Cooper)

A suposição de Retornos Variáveis de Escala (RVE) ou VRS (*Variable Return to Scale*) só é apropriada quando todas as DMUs estão operando em uma escala ótima. A competição imperfeita, restrições de capital, regulamentações governamentais, etc. podem fazer com que as firmas não operem em uma escala ótima [COELLI 1996]. Através desta máxima, Banker, Charnes e Cooper sugeriram, em 1984, um ajuste ao modelo CCR. Desta forma, tal modelo ficou, também, conhecido por BCC, que são as iniciais dos criadores do modelo. Esse modelo pressupõe RVE; ou seja, pode possuir, de acordo com Enzo Mariano et. al. (2006), os três tipos de retorno de escala:

- **Retorno Crescente (RNC) ou Não-Decrescente** – quando o aumento do número de entradas ocasiona o aumento desproporcional maior no número de saídas, o que ocorre quando uma firma está operando muito abaixo de sua capacidade ótima.
- **Retorno Constante** – quando o aumento do número de entradas causa um aumento proporcional nas saídas, ou seja, uma firma com retorno constante, operando em sua capacidade ótima.
- **Retorno Decrescente (RND) ou Não-Crescente** - quando o aumento do número de entradas, ocasiona o aumento desproporcional menor no número de saídas; o que ocorre, quando uma firma está operando muito acima de sua capacidade ótima.

Com a presença de vários tipos de retorno, as restrições impostas aos modelos de multiplicadores e de envoltório mudam. Assim, quando é RCE $\mu_o=0$, quando RVE μ_o é livre, quando o RNC $\mu_o \leq 0$ e quando o RND $\mu_o \geq 0$.

Em concomitância ao modelo CCR, o modelo BCC também possui as formas: fracionária, dos Multiplicadores e de Envoltório. Outra característica marcante do modelo BCC é que este geralmente apresenta valores maiores que os do modelo CCR. Com isso, surge o conceito de eficiência de escala [FERREIRA e GOMES 2009], que resulta do nível de eficiência máxima mais adequada, em razão da tecnologia a ser adotada. Tal situação é chamada de Tamanho de Escala Mais Produtiva ou MPSS (*Most Productive Scale Size*). A eficiência de escala é calculada pela fórmula:

$$EFE = \frac{qCCR}{qBCC} \quad (20)$$

Outra diferença significativa do BCC, em relação ao CCR, é que a fronteira de eficiência, gerada pelo modelo BCC, é convexa, curva e não-radial. Para uma ilustração mais clara, observe a figura a seguir que pressupõe uma fronteira pontilhada linear, do modelo CCR, e uma fronteira convexa, curva e não radial, referente ao modelo BCC.

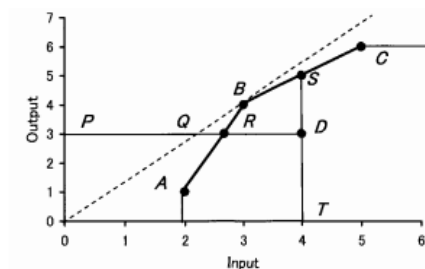


Figura 2.4. Fronteiras CCR x BCC [COOPER et. al. 2007].

As fórmulas para cálculo de eficiência técnica do modelo BCC, na forma Primal (Multiplicadores) e Dual (Envoltório), são apresentadas a seguir.

Modelo dos Multiplicadores – Primal

Orientação a Insumo:

$$\text{Maximizar a Eficiência} = \sum_{j=1}^m m_j y_{io} + m_o \quad (21)$$

Sujeito a:

$$\sum_{i=1}^j v_i x_{io} = 1 \quad (22)$$

$$\sum_{j=1}^m m_j l_{jk} - \sum_{i=1}^t v_i x_{ik} + m_o \leq 0 \quad (23)$$

$$m_j, v_i \geq 0$$

Orientação a Produto:

$$\text{Minimizar Eficiência} = \sum_{i=1}^r v_i x_{io} + v_o \quad (24)$$

Sujeito a:

$$\sum_{j=1}^s m_j l_{jo} = 1 \quad (25)$$

$$\sum_{j=1}^s m_j l_{jk} - \sum_{i=1}^r v_i x_{ik} + v_o \leq 0 \quad (26)$$

$$m_j, v_i \geq 0$$

Modelo dos Envoltórios – Dual

Orientação a Insumo:

$$\text{Minimizar Eficiência} = q$$

Sujeito a:

$$q x_{io} - \sum_{k=1}^n l_k x_{ik} \geq 0 \quad (27)$$

$$\sum_{k=1}^n l_k y_{mk} - y_{mo} \geq 0 \quad (28)$$

$$l_k \geq 0 \quad (29)$$

Orientação a Produto:

$$\text{Minimizar Eficiência} = f \quad (30)$$

Sujeito a:

$$x_{io} - \sum_{k=1}^n x_{ik} l_k \geq 0 \quad (31)$$

$$\sum_{k=1}^n y_{mk} l_k - f y_{mo} \geq 0 \quad (32)$$

$$l_k \geq 0$$

Como no CCR, o BCC produzirá resultados idênticos, nos modelos Primais e Duais, em ambas as orientações. Além disso, algumas destas fórmulas serão preenchidas com os dados das DMUs reais, utilizadas neste trabalho, para que se possa calcular a eficiência técnica relativa à DMU num software de resolução de problemas de programação linear chamado LINDO.

2.4.3. Modelo Aditivo de Folgas ou Modelo Baseado nas Folgas (MBF)

O Modelo Aditivo de Folgas foi proposto por Charnes et. al. (1985) e, também, é conhecido como Medida de Ajuste de Escala ou RAM (Range-Adjusted Measure), ou Modelo Baseado nas Folgas (MBF). Este modelo é importante, pois segundo Ferreira e Gomes (2009) ele considera, ao mesmo tempo, tanto a possibilidade de redução dos insumos, quanto o aumento da produção, mas baseado nas folgas dos insumos e produtos.

O MBF inclui folgas diferentes de zero que dão uma estimativa de excessos de entrada e deficiências nas saídas que poderiam ser melhoradas sem a perda de qualquer entrada ou saída [LEVERTY e GRACE 2007].

Novamente o trabalho de Ferreira e Gomes (2009) mostra que o MBF pressupõe que o valor marginal das folgas das entradas e saídas maiores que zero seja igual. Assim, deve-se ter cuidado ao empregar tal modelo, pois: a) as unidades de medida, usadas para as entradas e saídas influenciam nos resultados, i. e., unidades de medida diferentes resultam em medidas de eficiência diferentes; b) é necessário ter cuidado ao somar medidas que não sejam comensuráveis. Portanto, é muito recomendado que, quando usar a DEA, tenha-se bastante experiência com o setor ou organização em análise.

Além de todas estas suposições, o MBF se dá ao luxo de introduzir pesos nas unidades de entrada saída para agir no modelo, de acordo com o julgamento de especialistas. Estes pesos possuem o viés da pessoalidade, ou seja; para um especialista uma unidade pode possuir um determinado peso, mas para outro especialista já pode ser outro peso.

O MBF é importante, pois uma empresa pode ter eficiência técnica de 100%, mas mesmo assim, pode possuir excessos de insumos ou falta de produtos. Essas empresas são chamadas de falso eficientes. Porém, tal modelo ajuda à(s) empresa(s) analisada(s) a se tornar(em) realmente eficiente(s).

Diferentemente dos modelos CCR e BCC, o MBF só possui as formas de envoltório, sem nenhuma orientação, como pode ser visto nas fórmulas abaixo.

Modelo Aditivo ou Modelo Baseado em Folgas

$$\text{Maximizar } \sum_{i=1}^r s_i^- + \sum_{m=1}^s s_m^+ \quad (33)$$

$$(l, s_{ik}^-, s_{mk}^+)$$

Sujeito a:

$$\sum_{k=1}^n l_k x_{ik} + s_i^- = q x_{io} \quad (34)$$

$$\sum_{k=1}^n l_k y_{mk} - s_m^+ = y_{mo} \quad (35)$$

$$s_i^-, s_m^+, l_k \geq 0 \quad (36)$$

2.4.4 Outros modelos da DEA

A DEA vem despertado o interesse de vários pesquisadores no mundo inteiro, por isso vários destes proporam novos modelos para o DEA. Abaixo, estão listados alguns destes modelos com uma breve descrição:

- **Modelo de Medidas Específicas (MME):** variáveis não discricionárias, não controláveis, exógenas ou fixas. Este modelo trabalha com variáveis que estão fora do controle dos tomadores de decisão, e são chamadas de variáveis discricionárias, exógenas ou fixas,

em que os tomadores de decisão não podem manipular seus valores. Desta forma, o MME leva em conta tanto as variáveis discricionárias, quanto às não discricionárias.

- Modelo de Supereficiência (MSE). Na DEA original, quando várias empresas possuem o escore de eficiência igual a 1, não se pode distinguir hierarquicamente quais são as DMUs mais ou menos eficientes. Deste modo, o MSE faz este ranqueamento.
- Modelo FDH (*Free Disposal Hull*). Foi proposto por Cherchye et. al. (2000) e sugere o relaxamento da restrição da convexidade, quando é empregado um modelo de programação linear inteira mista, com a criação de uma fronteira eficiente de possibilidades, em forma de escada, com variações discretas.
- Modelo de variáveis categóricas ou *Dummies*. É semelhante ao MME, diferindo, apenas por empregar variações temporais nas variáveis.
- Entre outros.

Com o passar do tempo, novos modelos da DEA vão surgindo. Uma fonte atualizada com vários modelos da DEA pode ser encontrada, no sítio DEAZONE.

2.5 Comentários

Este capítulo apresentou todo um arcabouço teórico que suportará o alcance do objetivo deste trabalho. Assim, foi apresentado um histórico da programação linear, junto com alguns de seus fundamentos e o funcionamento do algoritmo Simplex. Em seguida, foi mostrado um histórico da DEA, passando pelos conceitos básicos; seus modelos, com seus formalismos e orientações, além de outros modelos complementares.

Capítulo 3

Este capítulo trata da descrição do levantamento dos dados, utilizados para estimar a fronteira de eficiência com a DEA. Ainda é mostrada uma filtragem nos dados, obtidos pelo método da pesquisa, através de formulários entregues aos ISPs, denominado análise de correlação. Este, ainda apresenta uma descrição detalhada da metodologia empregada, além da arquitetura proposta neste trabalho. A segunda parte do capítulo trata da normalização dos dados colhidos através da matriz de correlação. A terceira parte é remetida à aplicação dos dados, normalizados às formulas da DEA, segundo seus modelos, formas e orientações no software LINDO; além dos cálculos dos alvos (valores a serem atingidos pelas DMUs, nas entradas e saídas, com o intuito de que estas sejam produtivas e eficazes), movimentos radiais, folgas (excessos) das entradas e deficiências (faltas) das saídas.

3.1. Cenário

Os dados foram coletados através de formulários a serem preenchidos pelos responsáveis e/ou proprietários dos ISPs, localizados nas cidades do sul do interior (cariri) do Ceará, mais especificamente nas cidades de Crato e Juazeiro do Norte, além de alguns ISPs, localizados na cidade de Cajazeiras no alto-sertão da Paraíba.

3.1.1. Dados Reais

A avaliação dos ISPs se deu, através de questionários entregues às referidas empresas que as responderam com dados estimativos em cidades do interior do Ceará e Paraíba, cidades estas dispostas na Figura 3.1. O questionário, inicialmente, continha três perguntas, relacionadas às entradas/insumos (Planos Vendidos, Número de Clientes por Plano e a Velocidade Média Registrada por Cliente por Plano); e três perguntas referentes às saídas/produtos (Número de Servidores de Rede, *Link* Contratado e a Estimativa de Lucro Mensal).



Figura 3.1. Mapa dos estados do Ceará e Paraíba com as cidades dos ISPs analisados.

Ao serem aplicados os questionários, observou-se que alguns dos ISPs não ofereciam um número de planos que outros concorrentes ofereciam. Ainda surgiu a possibilidade de se lidar com ISPs que não trabalhavam com tal vazão ou velocidade final para com o cliente. Logo, foi aplicada uma normalização para adequação dos dados das firmas em questão. Cabe, como observação, o fato de que alguns ISPs solicitaram que seus nomes reais não fossem divulgados.

Antes de serem avaliadas as DMUs, as entradas e as saídas passaram por uma matriz de correlação que, segundo Reinaldo et. al.. (2002), serve para medir e avaliar o grau de relação existente entre duas variáveis (x,y), através da disposição destes pontos em torno de uma reta. Este coeficiente de correlação varia de -1 a 1. Assim, a eliminação de variáveis se dá através de uma forte correlação, ou seja, que a relação entre os pontos seja maior que 0,85. Com o emprego da matriz de correlação, nas entradas e saídas originais, foi vislumbrado que algumas variáveis de entrada deveriam ser excluídas, com o intuito de dar mais veracidade na avaliação das DMUs.

Os dados originais, recolhidos pelos questionários estão dispostos na tabela 3.1 com as entradas; e na tabela 3.2, com as saídas. A matriz de correlação dos dados originais será mostrada na tabela 3.3.

Tabela 3.1. Entradas com os dados originais dos ISPs analisados

DMU	Entradas			
	Planos/Usuários			Planos/Média de Vazão
	128	256		128 256
M. Net	30	148		120 205
Baydenet	1000	1000		110 200
ADL Link	150	300		90 154
NetLine	420	1000		120 250
CePlug	60	200		115 200
HotLink	80	180		120 220
Cz.Net	50	150		100 190
PervUbiq	100	250		125 225
Throughput	50	180		120 220

Tabela 3.2. Saídas com os dados originais dos ISPs analisados

DMU	Saídas		
	Servidores	Link Contratado (Mbps)	Lucro Mensal
M. Net	2	2	R\$ 1.400,00
Baydenet	5	50	R\$ 50.000,00
ADL Link	5	30	R\$ 10.000,00
NetLine	2	125	R\$ 40.000,00
CePlug	4	20	R\$ 15.000,00
HotLink	2	30	R\$ 20.000,00
Cz.Net	2	15	R\$ 10.000,00
PervUbiq	3	50	R\$ 35.000,00
Throughput	2	30	R\$ 20.000,00

Tabela 3.3. Matriz de correlação dos dados originais.

	X1	X2	X3	X4	Y1	Y2	Y3
X1	1	-	-	-	-	-	-
X2	0,89	1	-	-	-	-	-
X3	-0,08	0,03	1	-	-	-	-
X4	0,07	0,32	0,87	1	-	-	-
Y1	0,51	0,3	-0,54	-0,63	1	-	-
Y2	0,46	0,79	0,24	0,61	-0,06	1	-
Y3	0,8	0,82	0,3	0,49	0,26	0,73	1

Os dados normalizados serão apresentados ainda neste capítulo. Como pode ser visto, na tabela acima, existem variáveis com o fator de correlação superior a 0,85; ou seja, deverão ser removidas, para dar mais acurácia à avaliação realizada pela DEA.

3.2. Metodologia

Este trabalho utiliza DEA ou Análise Envoltória de Dados, com o objetivo de se estimar uma fronteira Pareto-eficiente das DMUs ou firmas ou empresas analisadas que, no caso, foram os ISPs do cariri cearense e alto-sertão paraibano.

Conforme apresentado no capítulo 2, a DEA é um método econômico não-paramétrico de estimação de fronteira Pareto-Koopmans eficiente, tendo como premissa a avaliação de

unidades de tomada de decisão ou DMUs, ou firmas, entre outros. O pressuposto parte de que, para se produzir um determinado produto, precisa-se da transformação de dada quantidade de insumos em produtos. Como visto na DEA, as unidades não precisam, necessariamente, ser iguais, daí uma grande vantagem da DEA sobre outras metodologias.

A DEA, por sua vez, possui vários estágios para o cálculo da fronteira eficiente, que são:

1. Primeiro estágio – é aplicado o modelo original da DEA e suas respectivas orientações e formas, i.e., é usado o modelo CCR, na forma dos multiplicadores e de envoltórios, orientados a entradas e as saídas, onde as folgas são calculadas residualmente.
2. Segundo estágio – de acordo com [FERREIRA e GOMES 2009] consiste em maximizar a soma das folgas, pois se torna necessário determinar possíveis folgas diferentes de zero, após o primeiro estágio aplicado, ou seja, determinar se uma dada DMU é ou não falso eficiente em qualquer orientação. Um modelo de segundo estágio é o BCC em suas formas e orientações.
3. Múltiplos estágios – de acordo com Coelli (1996), esta abordagem demanda mais desperdício de esforço computacional, mas em compensação identifica os pontos eficientes tanto de entradas, quanto de saídas, i.e., nas DMUs ineficientes e falso eficientes as várias execuções dos problemas de programação linear levariam a determinação dos pontos eficientes, fazendo com que as DMUs em questão tornassem eficientes. Um modelo de múltiplos estágios é o Modelo Aditivo de Folgas ou MBF ou Modelo Baseado em Folgas.

Todos estes estágios foram utilizados por este trabalho com o intuito de, não só projetar saber quais das empresas avaliadas seriam eficientes, mas, também, para se saber identificar algumas DMUs, consideradas falso eficientes, que são aquelas que possuem folgas diferentes de zero e, assim, fazer com que todas as empresas avaliadas possam se tornar eficientes.

Além disto, este trabalho visa a não só aumentar a satisfação do cliente que contrata os serviços prestados pelos ISPs, mas também visa a aumentar a lucratividade dos ISPs, atuando, assim, em ambas as vertentes da DEA, i.e.; tanto na orientação a insumos, como na orientação aos produtos.

3.2.1. Arquitetura proposta

A arquitetura proposta por este trabalho visa, primeiramente, à coleta das informações julgadas primordiais para a solução dos problemas de programação linear, sobre os vários modelos, formas e orientações da DEA, em seus vários estágios.

Tais informações foram coletadas com base em formulários dirigidos, ou seja, as dúvidas dos entrevistados foram minimizadas com a presença do autor deste trabalho, para que as perguntas fossem respondidas a contento, no intuito de uma avaliação melhorada por parte da DEA.

Assim o **levantamento dos dados**, junto às empresas de provimento de serviços de conexão para com a Internet, faz parte da área externa da arquitetura, sendo uma das camadas mais importantes da mesma. Vale ser externado que muitas empresas, das localidades citadas, se negaram a responder o questionário, pois, ou desconhecem o caráter acadêmico, ou esperavam que o autor deste fosse mais um concorrente, atrás de informações estratégicas, para alavancar o seu negócio. A arquitetura pode ser vista na Figura 3.2 (abaixo).

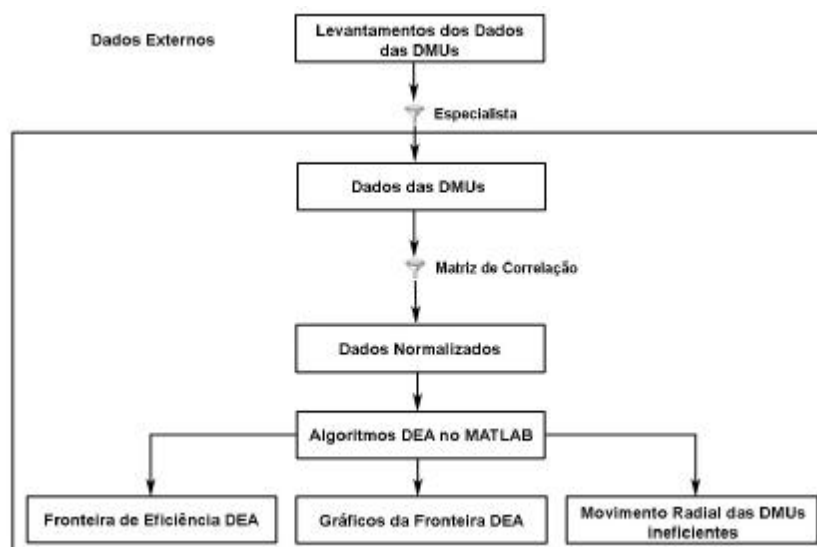


Figura 3.2. Arquitetura proposta pelo trabalho.

Antes de os dados serem inseridos nas camadas internas da arquitetura, foram filtrados por um especialista em redes de computadores para avaliar, primariamente, os dados recebidos dos ISPs, para a geração dos dados que farão parte da segunda camada da arquitetura, que são os dados das companhias a serem avaliadas. Estes dados foram mostrados nas tabelas 3.1 e 3.2; além do mais, vários ISPs não possuíam os planos ou velocidades que outros concorrentes ofereciam, daí outra razão para se efetuar tal normalização dos dados.

O segundo filtro aplicado é a **matriz de correlação** que analisa o grau de relacionamento que uma variável tem, comparado com outra no modelo; quais sejam, entrada com entrada, entrada com saída ou vice-versa. Neste caso, a tabela 3.3 mostrou uma matriz de correlação, onde as variáveis com fator de correlação acima de 0,85, que deveriam ser excluídos para aumentar a precisão da análise obtida com a DEA. Tal matriz de correlação será apresentada em seguida.

A terceira camada são os **dados normalizados**, obtidos através da matriz de correlação. Estes dados foram usados para obtenção da fronteira de eficiência, calculada pela DEA, e serão apresentados adiante.

A quarta camada é a camada da inteligência, i.e., onde são calculadas as fronteiras de eficiência, segundos os modelos, suas formas e orientações, nos primeiro, segundo e múltiplos estágios. Tal camada é referida aos **algoritmos da DEA**, desenvolvidos na linguagem matemática, baseada em matrizes MATLAB, que conta com uma série de funções matemáticas prontas; dentre elas, a linprog que resolve problemas de programação linear, através do algoritmo Simplex e que são obrigatórios, ao se usar a metodologia DEA, para a obtenção da fronteira de eficiência.

As últimas camadas são geradas pela quarta camada, como forma de apresentação ou saída dos dados obtidos, através da quarta camada, pois nestas camadas inferiores estão a **fronteira de eficiência** junto com as DMUs, que são consideradas como *benchmarks*; as ineficientes e as falso eficientes e suas respectivas folgas; os **gráficos de ranqueamento da fronteira DEA** e os **movimentos radiais**, para tornarem as DMUs ineficientes e falso eficientes em eficientes, seguindo seus respectivos parceiros de referência.

Da mesma forma que uma arquitetura foi proposta para embasar o trabalho foi sugerido um fluxograma algorítmico, a fim de mostrar a organização dos dados, usados nesta dissertação, que pode ser visto na Figura 3.3.

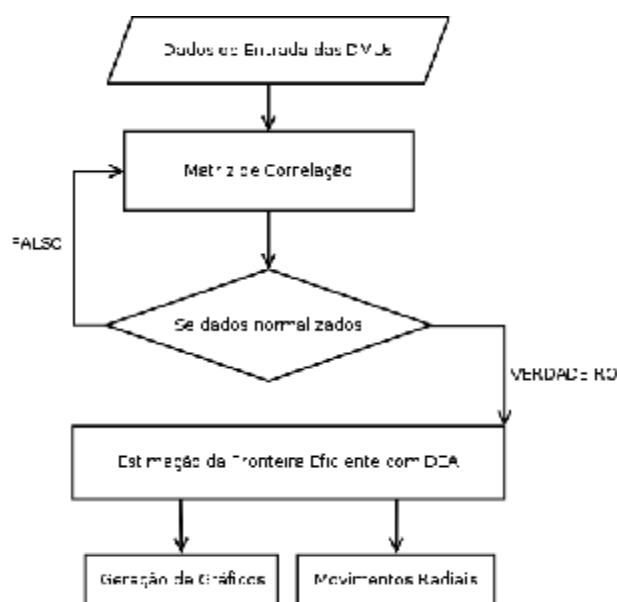


Figura 3.3. Fluxograma algorítmico para estimação de uma fronteira eficiente DEA.

3.3. Normalização dos Dados

Como citado anteriormente, os dados foram normalizados ou filtrados, a partir do cálculo de matrizes de correlação, calculadas a partir das variáveis de entrada e saída. A primeira normalização foi apresentada na tabela 3.3, onde foram excluídas as variáveis que apresentavam fator de correlação acima de 0,85.

A segunda, e definitiva normalização, calculada através do aplicativo de planilha eletrônica OpenCalc, da suíte de aplicativos OpenOffice, poderá ser apreciada na tabela 3.4, a seguir:

Tabela 3.4. Matriz de correlação definitiva dos dados das DMUs.

	X1	X2	Y1	Y2	Y3
X1	1	-	-	-	-
X2	0,35	1	-	-	-
Y1	0,45	-0,45	1	-	-
Y2	0,59	0,65	-0,01	1	-
Y3	0,82	0,49	0,35	0,72	1

Depois destas normalizações ou filtragens realizadas, os dados estão prontos a serem utilizados para calcular a fronteira de eficiência com a DEA, podendo ser visualizados na tabela abaixo.

Tabela 3.5. Dados das DMUs normalizados.

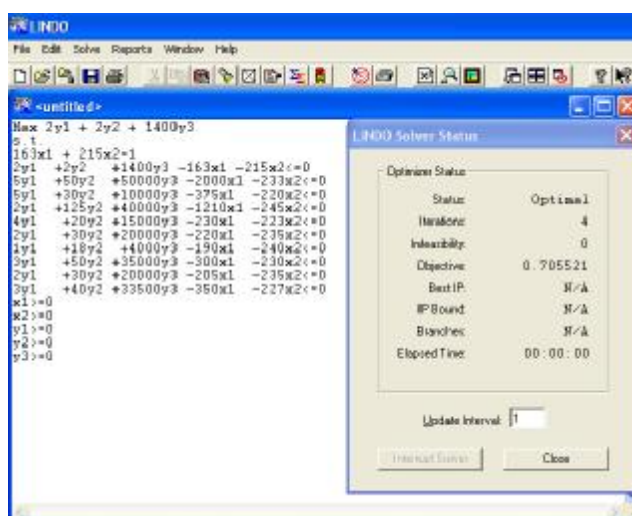
DMU	X1	X2	Y1	Y2	Y3
M. Net	163	215	2	2	R\$ 1.400,00
Baydenet	2000	233	5	50	R\$ 50.000,00
ADL Link	375	220	5	30	R\$ 10.000,00
NetLine	1210	245	2	125	R\$ 40.000,00
CePlug	230	223	4	20	R\$ 15.000,00
HotLink	220	235	2	30	R\$ 20.000,00
Cz.Net	190	240	1	18	R\$ 4.000,00
WiNET	300	230	3	50	R\$ 35.000,00
Throughput	205	235	2	30	R\$ 20.000,00

Onde as entradas são: X1, que é o número de usuários que usam 256 kbps; X2 é a velocidade média registrada pelos clientes. E as saídas são: Y1, que é o número de servidores de rede utilizados pelo provedor de Internet, na localidade pesquisada; Y2 é o *link* contratado pelo ISP e Y3 é a estimativa de lucro mensal.

3.3 Dados normalizados na DEA com seus respectivos cálculos

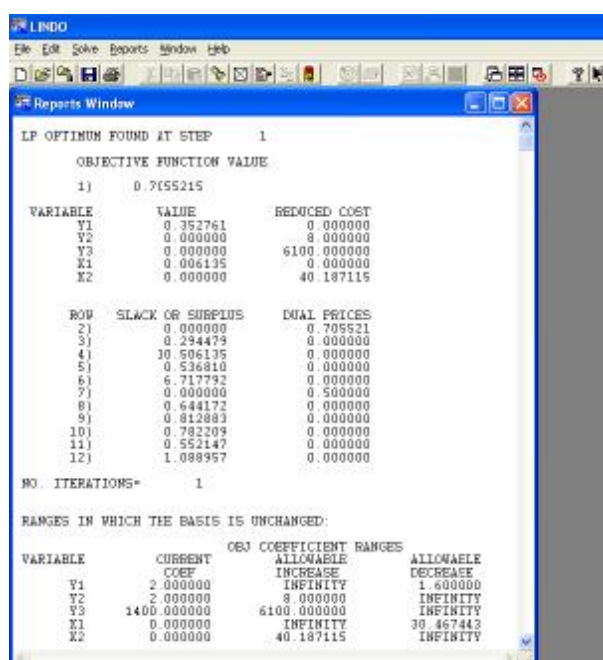
Esta seção foi criada para facilitar o entendimento dos formalismos da DEA, para se estimar as fronteiras de eficiência e, ao mesmo tempo, mostrar a aplicação das fórmulas, apresentadas no capítulo 2, usando o aplicativo LINDO, que calcula valores ótimos para as variáveis através da programação linear, através do uso do método Simplex.

Seguindo a ordem cronológica da DEA será, primeiramente, mostrada uma figura que expõe como se calcular a eficiência de uma DMU. No caso, a M.Net da cidade de Crato-CE, segundo os dados da tabela 3.5 que contém os dados normalizados.

**Figura 3.4. Fórmulas do modelo CCR/M/I aplicados a uma das DMUs analisadas.**

As fórmulas utilizadas foram as (6), (7), (8) e (9) do capítulo 2, que são do modelo CCR, na forma primal ou dos multiplicadores e orientada a insumos ou a entrada. É importante notar que, quando o LINDO soluciona (*solve*) problemas de programação linear, ele exibe uma tela de *status* com um sumário resumido dos cálculos efetuados. Entretanto, o LINDO exibe informações mais detalhadas da resolução dos cálculos, caso seja de interesse do usuário. Assim, tal software exibe estas informações tal como na figura adiante.

É importante notar a rapidez no desempenho computacional, ao se encontrar a solução ótima, através do algoritmo Simplex, ao ser vislumbrado o número de iterações (*iterations*) ou repetições necessárias para se encontrar uma solução ótima no espaço de possibilidades.



LP OPTIMUM FOUND AT STEP 1

OBJECTIVE FUNCTION VALUE

1) 0.7055215

VARIABLE	VALUE	REDUCED COST
Y1	0.352761	0.000000
Y2	0.000000	8.000000
Y3	0.000000	6100.000000
X1	0.006135	0.000000
X2	0.000000	40.187115

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.705521
3)	0.294479	0.000000
4)	10.506135	0.000000
5)	0.536810	0.000000
6)	6.717792	0.000000
7)	0.000000	0.500000
8)	0.644172	0.000000
9)	0.812883	0.000000
10)	0.782209	0.000000
11)	0.552147	0.000000
12)	1.088957	0.000000

NO. ITERATIONS= 1

RANGES IN WHICH THE BASIS IS UNCHANGED:

VARIABLE	CURRENT COEF	ALLOWABLE INCREASE	ALLOWABLE DECREASE
Y1	2.000000	INFINITY	1.600000
Y2	2.000000	8.000000	INFINITY
Y3	1400.000000	6100.000000	INFINITY
X1	0.000000	INFINITY	30.467443
X2	0.000000	40.187115	INFINITY

Figura 3.5. Relatório da resolução de problemas de programação linear com o LINDO.

Para se gerar a fronteira de eficiência com a DEA, é preciso repetir tais fórmulas para todas as DMUs a serem analisadas, em todos os modelos, formas e orientações. Contudo, os resultados obtidos no mesmo modelo serão idênticos sob todas as formas e orientações; mas, como exposto anteriormente, as eficiências técnicas do modelo BCC são, geralmente, maiores do que no modelo BCC, onde razão entre a eficiência técnica do modelo CCR pela eficiência técnica do modelo BCC resulta na eficiência de escala (observe a fórmula (20) do capítulo 2).

Em seguida, será mostrado como se calcular a eficiência técnica da DMU, chamada HotLink de Cajazeiras –PB com o LINDO, através do modelo dos multiplicadores, orientação a produtos ou saídas que foram obtidos usando as fórmulas (10), (11) e (12) do capítulo 2.

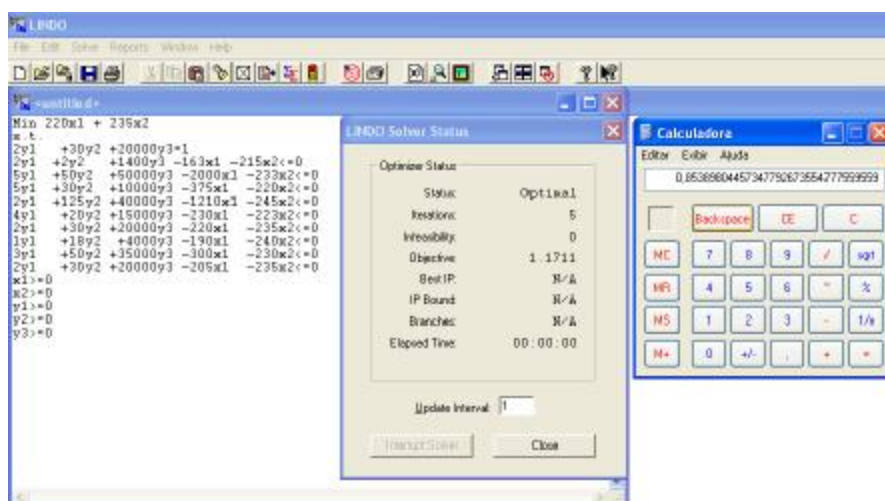


Figura 3.6. Fórmulas do modelo CCR/M/O aplicados a uma das DMUs analisadas.

Relembrando que o resultado da eficiência técnica q será obtido pela divisão entre a unidade (ou 1) por $f = 1.1711$ que, no caso, será igual a 0.8538, como pode ser visto na figura acima.

Abaixo, será ilustrado como se calcular a eficiência técnica, segundo o modelo CCR/E/I. Para fins de prova, a DMU a ser analisada será, novamente, a empresa M. Net, mostrando que os resultados obtidos, através do modelo CCR/M/I; bem como CCR/M/O, serão os mesmos, bastando somente este exemplo como prova, não necessitando da apresentação do modelo CCR/E/O.

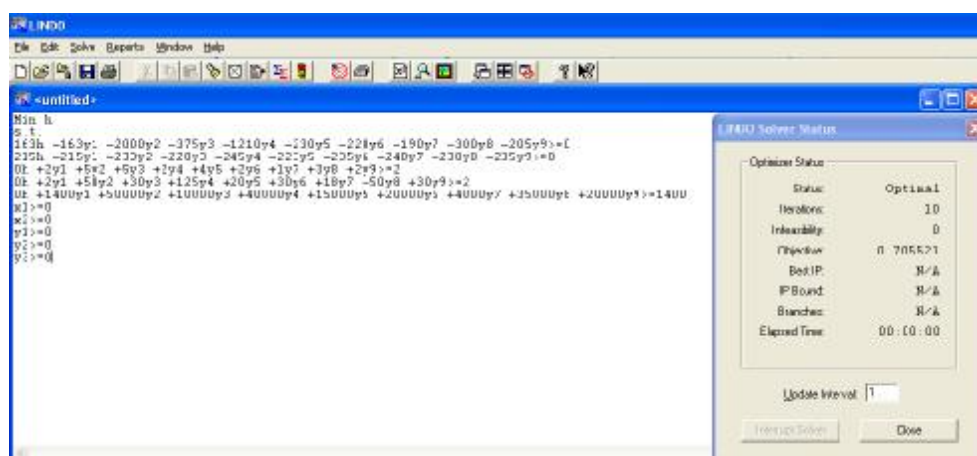


Figura 3.7. Fórmulas do modelo CCR/E/I aplicados a uma das DMUs analisadas.

Os resultados obtidos da DMU M. Net foram obtidos com o uso das fórmulas (13), (14) e (15) do capítulo 2. Uma observação deve ser feita com o uso destas fórmulas, no LINDO, pois pode ser visto que o número de fórmulas é bem menor do que no modelo primal ou de multiplicadores, i.e., no modelo primal o número de inequações ou equações é o número de

DMUs, sendo que, na forma dual ou dos envoltórios (envoltório), é reduzido para o número de variáveis em estudo no modelo.

Para se calcular as folgas e as insuficiências dos produtos em múltiplos estágios, é necessário já possuir a eficiência técnica de uma determinada DMU, a fim de poder multiplicar o valor da eficiência técnica tanto pela entrada, quanto pela saída, obtendo os movimentos radiais em direção à fronteira. Com as soluções dos problemas de programação linear, bem como ao serem visualizadas as fórmulas da DEA, segundo seus modelos, formas e orientações, nota-se a existência de uma variável λ , variável esta que determina os parceiros de referência entre as DMUs, sendo que, com estes valores, são selecionados quais valores de outras DMUs poderão ser usados nos cálculos das folgas e das insuficiências de produtos. De posse desta(s) informação(ões) são calculados os alvos (*targets*) de cada uma das DMUs, assim a diferença entre o movimento radial de cada entrada/saída e os alvos de cada entrada/saída determinará as folgas e as deficiências.

Como exemplo a DMU HotLink de Juazeiro do Norte-CE, com os λ s possuem os seguintes valores: 0,0714 e 0,5714 que correspondem, respectivamente, aos seus parceiros CEPlug e Throughput. Desta forma, os valores destas DMUs deverão ser aplicados para se encontrar os valores tanto dos alvos que, em seguida, serão usados para serem encontradas tanto as folgas, quanto das faltas; além de serem selecionados na matriz que forma as DMUs, somente as linhas e colunas que correspondem aos insumos a serem calculados os alvos. Assim, para se calcular os alvos da DMU supracitada observe as equações abaixo:

Para as entradas:

$$\sum_{i=1}^k l_k x_k, \forall l > 0 \quad (1)$$

$$\sum_{i=1}^k l_k y_k, \forall l > 0 \quad (2)$$

Vale ser observado que o valor de λ tem de ser maior que zero, desta forma a equação (1) calcula os alvos das entradas e a (2), os das saídas. Como exemplo de aplicação das fórmulas acima está o exemplo do cálculo dos alvos de todas as entradas e saídas da DMU HotLink.

$$Alvo_X1 = (0,0714 \times 230) + (0,5714 \times 300)$$

$$Alvo_X2 = (0,0714 \times 223) + (0,5714 \times 230)$$

$$Alvo_Y1 = (0,0714 \times 4) + (0,5714 \times 3)$$

$$Alvo_Y2 = (0,0714 \times 20) + (0,5714 \times 50)$$

$$Alvo_Y3 = (0,0714 \times 20000) + (0,5714 \times 35000)$$

No caso da DMU HotLink de posse do alvo, o cálculo do movimento radial é obtido, através da multiplicação da eficiência técnica (q) da DMU pela entrada/saída em questão. Como a eficiência técnica da DMU HotLink, segundo o CCR é 0,8539, então os valores dos movimentos radiais desta DMU podem ser obtidos através dos cálculos abaixo:

$$Radial_X1 = 0,8539 \times 220$$

$$Radial_X2 = 0,8539 \times 235$$

$$Radial_Y1 = 0,8539 \times 2$$

$$Radial_Y2 = 0,8539 \times 30$$

$$Radial_Y3 = 0,8539 \times 20000$$

Finalmente, para serem calculadas as folgas e as deficiências basta subtrair, para cada entrada e saída, o movimento radial pelo alvo em questão, podendo ser visualizado, em apenas um exemplo a seguir, no caso para se calcular a folga da primeira entrada (X1) e falta (deficiência) da primeira saída (Y1).

$$Folga_X1 = Radial_X1 - Alvo_X1$$

$$Falta_X1 = Radial_Y1 - Alvo_Y1$$

Enfim, será apresentado o modelo aditivo ou MBF, que, também, como os outros, pode ser resolvido usando o LINDO. Uma peculiaridade, acerca deste modelo, é que ele só possui a forma dual, por conseguinte este modelo também não possui nenhuma orientação, de entrada ou saída. Assim, tal modelo pode ser resolvido através da figura a seguir. Este modelo buscará por falso eficientes, tentando, ao mesmo tempo, diminuir as entradas e maximizar as saídas, buscando os movimentos radiais, visando a eliminação dos excessos dos insumos e as folgas dos produtos.

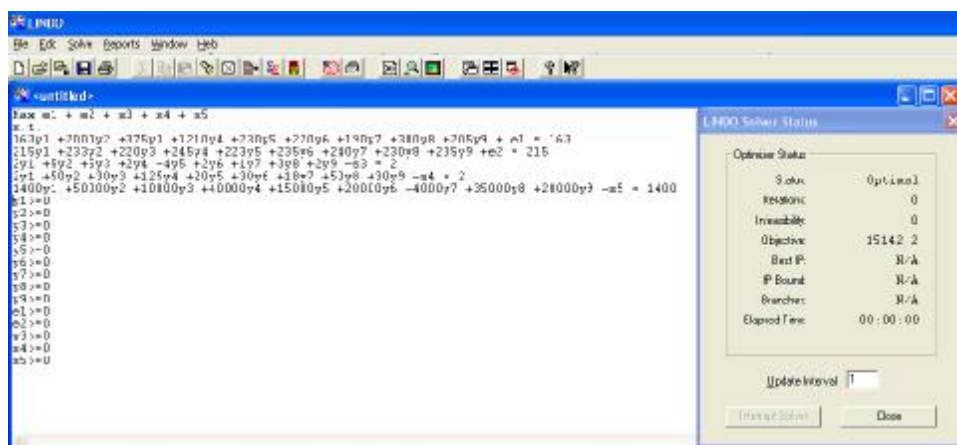


Figura 3.8. Fórmulas do modelo MBF aplicados a uma das DMUs analisadas.

Vale ser observado que o valor da função objetivo é 15142.2, ou seja, o valor da soma das folgas (excessos) e das insuficiências de saídas é o valor acima. Para serem obtidos tais valores, foram usadas as fórmulas (33), (34), (35) e (36) do capítulo 2.

Para esclarecer o valor da função objetivo, faz-se necessário a visualização dos dados de relatórios do LINDO.

The screenshot shows the LINDO Reports Window. The table displays the objective function value and the reduced cost for each variable. The objective function value is 15142.23. The reduced cost for each variable is shown in the third column.

Variable	Value	Reduced Cost
E1	0.00000	185.056870
E2	79.87647	0.00000
S1	0.00000	6998.019631
S2	21.17647	0.00000
S3	15041.17675	0.00000
T1	0.00000	15142.239492
T2	0.00000	207381.625000
T3	0.00000	24966.224609
T4	0.00000	-7.250.75625
T5	3.217647	0.00000
T6	0.00000	7119.470707
T7	0.00000	24573.736156
T8	3.376473	0.00000
T9	0.00000	4348.617676

Figura 3.9. Relatório das folgas e faltas do MBF relativo a uma das DMUs analisadas.

É importante notar que, para a DMU M. Net, não existe folga (excesso) da entrada 1, mas existe folga da entrada 2 em 79.876. Em relação às saídas, não existe nenhuma falta no produto 1, mas existe uma falta de 21.176 do produto 2, além de uma falta de 15041.176 para o produto 3. Assim, a soma de todas estas folgas, com estas faltas, resultará no valor de 15142.23.

Outra informação importante é que estes valores são diferentes das folgas e deficiências, calculadas anteriormente, pelo próprio pressuposto básico da MBF apresentado anteriormente.

3.4 Comentários

Este capítulo tratou da descrição do cenário que foi utilizado para estimação de fronteira de eficiência avaliada pela DEA. Desta forma, foram apresentados os dados, na forma com que foram originalmente coletados e que, posteriormente, foram normalizados pela aplicação de uma matriz de correlação. Ainda foi apresentada a metodologia utilizada, no caso DEA, mostrando a arquitetura proposta pelo trabalho, além de um fluxograma algorítmico, utilizado para a confecção básica dos algoritmos que foram desenvolvidos para a realização de todos os cálculos do trabalho. Foram mostrados os dados normalizados pelo uso da técnica de matriz de correlação. Em seguida, estes dados normalizados foram incorporados às diversas fórmulas da DEA, segundo seus modelos, formas e orientações usadas neste trabalho.

Capítulo 4

Resultados e Discussões

Este capítulo apresenta, inicialmente, o ambiente de desenvolvimento onde foi concebido o *Solver* DEA proposto. Em seguida, serão apresentados os resultados, obtidos com a execução do *Solver* DEA, em seus principais modelos: CCR e BCC; formas: multiplicadores e envoltório e orientações: entrada e saída. As seções apresentarão as análises, confrontadas com o Win4DEAP, para mostrar a precisão nos resultados obtidos, além do que algumas análises não implementadas pelo *Solver* DEA proposto são apresentadas, bem como alguns gráficos serão mostrados. Na última seção, são mostrados os resultados obtidos pelo software proposto por este trabalho para análise das DMUs, segundo o MBF.

4.1. Ambiente de Desenvolvimento

O *Solver* DEA proposto foi desenvolvido no MATLAB (MATrix LABoratory) para calcular as fronteiras de eficiência nos principais modelos da DEA, tal como: CCR e BCC, com suas respectivas formas: dos Multiplicadores e de Envoltório, além das orientações de entrada e saída.

Como o MATLAB utiliza as matrizes, como tipo de dados padrão para realizar as operações matemáticas, este trabalho o julgou necessário para se calcular as eficiências das DMUs em análise. O MATLAB ainda possui uma série de funções matemáticas e de outros objetivos já prontas. Dentre estas funções há de se destacar a linprog que usa métodos de programação linear, em busca de valores ótimos, onde pode ser utilizado o método Simplex, usado neste trabalho e na DEA, além do método dos pontos interiores.

Além disso, o MATLAB conta com uma série de ferramentas de geração de gráficos e uma gama de outras funcionalidades, implementadas num pacote simulador, denominado Simulink que, dentre as ferramentas de simulação, podem ser citadas: a simulação mecânica, hidráulica, de potência de sistemas, de eventos e muitas outras. Outro atrativo do MATLAB é uma miríade de ferramentas de cálculo (*Toolboxes*) prontas, tais como: cálculo aeroespacial, lógica fuzzy, comunicações, redes neurais, processamento de imagens, bioinformática, etc.

Outra vantagem do uso do MATLAB é a interoperabilidade com os principais Sistemas Operacionais como Windows, Linux e MAC OS X. Entretanto, um programa desenvolvido neste pode ser traduzido para linguagem C, aumentando ainda mais a portabilidade de um sistema. Outro ponto importante é a capacidade de comunicação com outras linguagens de programação como JAVA e C++, além da orientação a objeto.

A implementação do *Solver* DEA proposto pode ser visualizada nos anexos da dissertação.

4.2. Resultados Obtidos

Nas subseções seguintes serão apresentados os resultados, obtidos pelo software proposto, utilizando DEA em seus principais modelos, formas e orientações.

4.2.1. Resultados das DMUs analisadas segundo o CCR

O CCR é o modelo padrão da DEA, pois foi o primeiro a ser desenvolvido por Charnes, Cooper e Rhodes (1978). Tal modelo é baseado na suposição dos Retornos Constantes de Escala onde as entradas e as saídas são proporcionais entre si. O CCR, originalmente, foi proposto para ser trabalhado, usando a forma fracionária, mas seus autores logo vislumbraram a redução da forma fracionária para um problema de programação linear e, assim, criaram os seus outros dois modelos: o de Multiplicadores (Primal) e de Envoltórios (Dual), ao usar o algoritmo Simplex, para encontrar as fronteiras de eficiência DEA.

Como visto no capítulo anterior, os dados das DMUs analisadas foram obtidos através de questionários, realizados nas empresas pesquisadas, onde nem todas as corporações vendiam os mesmos serviços, assim foram aplicadas as matrizes de correlação duas vezes para aumentar a precisão do julgamento da eficiência das DMUs analisadas. Entretanto, foi aplicada, também, uma normalização nos dados de entradas das DMUs, deixando apenas a velocidade de 256 kbps e o número de clientes deste plano como entradas, permanecendo as saídas inalteradas, pois as matrizes de correlação julgaram desta forma.

Os dados obtidos foram confrontados com o software Win4DEAP, desenvolvido como extensão do DEAP original proposto por Coelli (1996). O Win4DEAP nada mais é do que uma ferramenta visual do DEAP, assim tal software. Além do LINDO foram usados para confrontar a veracidade dos dados do *Solver* DEA proposto, sendo que os dados obtidos, nestas três plataformas, foram os mesmos, atestando, assim, que o *Solver* DEA proposto é realmente funcional para qualquer modelo proposto, em suas formas e orientações.

Como a principal preocupação deste trabalho é aumentar a satisfação dos clientes, este se preocupou, primariamente, em calcular os movimentos radiais, em direção à fronteira de eficiência, as folgas ou excessos das entradas e as deficiências ou faltas das saídas, além dos alvos eficientes no modelo CCR orientado a entradas. Os resultados obtidos serão mostrados abaixo da forma como é a saída original do *Solver* DEA proposto.

===== DEA MATLAB Solver =====																			
===== Eficiência =					RADIAL					= FOLGAS (SLACKS) ==					FALTAS ==				
= DMU == Theta =					I1	I2	O1	O2	O3	I1	I2	O1	O2	O3	I1	I2	O1	O2	O3
=====																			
= M.Net	== 0.7055	== 115.00	== 151.69	== 1.41	== 1.41	== 987.73	== 0	== 40.19	== -0	== 8	== 6100	== 115	== 111.5	== 2	== 10	== 7500	==	==	==
= Baydenet	== 1.0000	== 2000.00	== 233.00	== 5.00	== 50	== 50000	== 0	== 0	== -0	== -0	== -0	== 2000	== 233	== 5	== 50	== 50000	==	==	==
= ADLLink	== 1.0000	== 375.00	== 220.00	== 5.00	== 30	== 10000	== 0	== 0	== -0	== -0	== -0	== 375	== 220	== 5	== 30	== 10000	==	==	==
= NetLine	== 1.0000	== 1210.00	== 245.00	== 2.00	== 125	== 40000	== 0	== 0	== -0	== -0	== -0	== 1210	== 245	== 2	== 125	== 40000	==	==	==
= CEPlug	== 1.0000	== 230.00	== 223.00	== 4.00	== 20	== 15000	== 0	== 0	== -0	== -0	== -0	== 230	== 223	== 4	== 20	== 15000	==	==	==
= HotLink	== 0.8539	== 187.86	== 200.67	== 1.71	== 25.62	== 17077.92	== 0	== 53.31	== -0	== -0	== 1071.43	== 187.86	== 147.36	== 2	== 30	== 21071.43	==	==	==
= Cz.Net	== 0.5684	== 108.00	== 136.42	== 0.57	== 10.23	== 2273.68	== 0	== 53.62	== 0.08	== -0	== 8600	== 108	== 82.80	== 1.08	== 18	== 12600	==	==	==
= WiNet	== 1.0000	== 300.00	== 230.00	== 3.00	== 50	== 35000	== 0	== 0	== -0	== -0	== -0	== 300	== 230	== 3	== 50	== 35000	==	==	==
Throughput	== 0.9164	== 187.86	== 215.35	== 1.83	== 27.49	== 18327.53	== 0	== 67.99	== -0	== 0	== 1071.43	== 187.86	== 147.36	== 2	== 30	== 21071	==	==	==

Figura 4.1. Saída do software proposto para o CCR/E/I.

Vale ainda ressaltar que o software proposto está sob constantes melhorias para proporcionar resultados mais inteligíveis. Entretanto, pode-se observar que as DMUs: M.Net, HotLink, Cz.Net e Throughput não são eficientes, visto que necessitam diminuir, proporcionalmente, em 29,45%, 14,61%, 43,16% e 8,36%, respectivamente, suas entradas com o intuito de tornarem-se eficientes, através do movimento radial. Mesmo assim, estas diminuições proporcionais sobre estas DMUs podem acarretar que ainda continuem ineficientes, pois ainda possuem, ora folgas de entrada, ora escassez de saídas, fazendo com que o software proposto tenha de calcular estes valores com a finalidade de alcançar os alvos considerados ótimos.

Ainda pode ser observado que as empresas CEPlug e WiNET são exemplos de excelência, ou seja, um *benchmark* das empresas M.Net, HotLink, CzNet e Throughput. É notório afirmar que, quanto maior o valor positivo de λ (WiNET), mais importante é a DMU eficiente como parceiro de referência; além disso, quanto mais vezes uma DMU eficiente é identificada como parceiro de excelência para DMUs ineficientes, maior é sua importância entre as DMUs eficientes [FERREIRA e GOMES 2009].

=====						
==	DMU	==	Parceiro(s) de Excelência	==	Lambdas	==
=====						
==	1 - M.Net	==	5.00		==	0.50
==	2 - Baydenet	==	2.00		==	1.00
==	3 - ADLLink	==	3.00		==	1.00
==	4 - NetLine	==	4.00		==	1.00
==	5 - CEPlug	==	5.00		==	1.00
==	6 - HotLink	==	5.00	8.00	==	0.07 0.57
==	7 - CzNet	==	8.00		==	0.36
==	8 - WiNet	==	8.00		==	1.00
==	9 - Throughput	==	5.00	8.00	==	0.07 0.57
=====						

Figura 4.2. Parceiros de Referência (Benchmrks) das DMUs.

Como mostrado anteriormente, os resultados do software proposto foram comparados ao Win4DEAP. Assim, um relatório do Win4DEAP será mostrado abaixo para fins de comparação. Na figura 4.4, será apresentado o sumário de eficiência técnica; na figura 4.5, (a)

têm-se o sumário de folgas das entradas, na figura 4.5 (b) é mostrado o sumário de escassez das saídas; e a figura 4.5 apresenta o sumário dos alvos das entradas (a) e das saídas (b).

EFFICIENCY SUMMARY:

firm	te
1	0.706
2	1.000
3	1.000
4	1.000
5	1.000
6	0.854
7	1.000
8	1.000
9	0.916

Figura 4.3. Sumário de Eficiências Técnicas das DMUs calculadas no Win4DEAP.

SUMMARY OF INPUT SLACKS:

firm	input:	1	2
1		0.000	56.961
2		0.000	0.000
3		0.000	0.000
4		0.000	0.000
5		0.000	0.000
6		0.000	62.430
7		0.000	0.000
8		0.000	0.000
9		0.000	74.196

SUMMARY OF OUTPUT SLACKS:

firm	output:	1	2	3
1		0.000	11.339	8646.087
2		0.000	0.000	0.000
3		0.000	0.000	0.000
4		0.000	0.000	0.000
5		0.000	0.000	0.000
6		0.000	0.000	1254.753
7		0.000	0.000	0.000
8		0.000	0.000	0.000
9		0.000	0.000	1169.282

Figura 4.4. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas.

SUMMARY OF INPUT TARGETS:

firm	input:	1	2
1		163.000	158.039
2		2000.000	233.000
3		375.000	220.000
4		1210.000	245.000
5		230.000	223.000
6		220.000	172.570
7		190.000	240.000
8		300.000	230.000
9		205.000	160.804

SUMMARY OF OUTPUT TARGETS:

firm	output:	1	2	3
1		2.835	14.174	10630.435
2		5.000	50.000	50000.000
3		5.000	30.000	10000.000
4		2.000	125.000	40000.000
5		4.000	20.000	15000.000
6		2.342	35.133	24676.806
7		1.000	18.000	40000.000
8		3.000	50.000	35000.000
9		2.183	32.738	22994.297

Figura 4.5. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas.

Como pôde ser visto, os resultados do Solver DEA proposto, comparados com os resultados do Win4DEAP, são os mesmos para o CCR/E/I.

Os gráficos, gerados pelo *Solver* DEA proposto, apenas mostra as eficiências técnicas das DMUs e pode ser visto na figura abaixo.

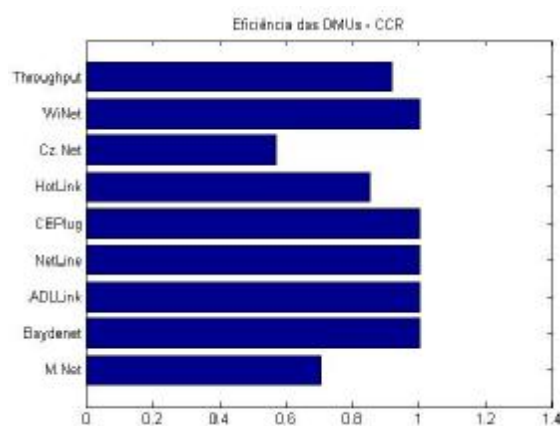


Figura 4.6. Gráfico das Eficiências Técnicas do modelo CCR gerado pelo Solver DEA proposto.

Foi utilizado o DEA-Solver da Saitech, uma versão acadêmica, desenvolvida na planilha eletrônica Excel, utilizando macros para ranquear as DMUs analisadas, além da geração de um gráfico com as eficiências técnicas do modelo CCR que serão mostrados na tabela e figura a seguir.

Tabela 4.1. Ranqueamento das DMUs analisadas segundo CCR/E/I usando o DEA-Solver da Saitech.

Rank	DMU	Score
1	WiNET	1
1	CePlug	1
1	Baydenet	1
1	ADL Link	1
1	NetLine	1
6	Throughpu	0,916376
7	HotLink	0,853896
8	M. Net	0,705521
9	Cz.Net	0,568421

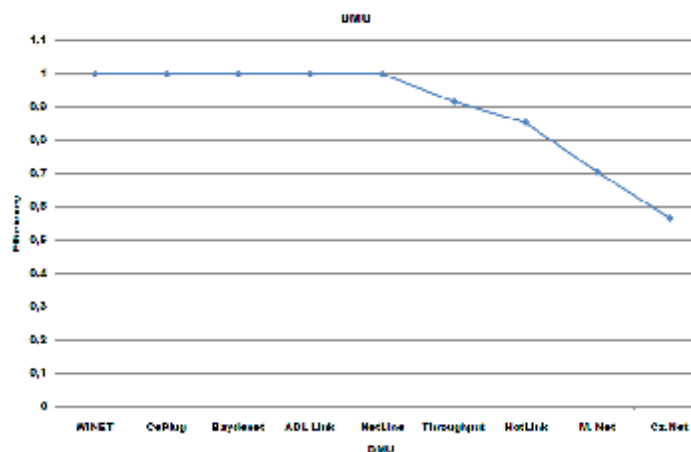


Figura 4.7. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.

As folgas de entradas, faltas das saídas e os alvos, usando o CCR/E/O, foram calculados usando o Win4DEAP, pois o *Solver* DEA proposto só calcula as eficiências técnicas deste modelo. Assim, os resultados são sumarizados nas figuras a seguir, sendo que as eficiências técnicas não serão mostradas, pois são as mesmas mostradas anteriormente. Apesar das eficiências técnicas serem idênticas, as folgas, faltas e alvos são diferentes do modelo CCR/E/I.

SUMMARY OF OUTPUT SLACKS:				SUMMARY OF INPUT SLACKS:				
firm	output:	1	2	3	firm	input:	1	2
1		0.000	11.339	8646.087	1		0.000	56.961
2		0.000	0.000	0.000	2		0.000	0.000
3		0.000	0.000	0.000	3		0.000	0.000
4		0.000	0.000	0.000	4		0.000	0.000
5		0.000	0.000	0.000	5		0.000	0.000
6		0.000	0.000	1254.753	6		0.000	62.430
7		0.000	0.000	0.000	7		0.000	0.000
8		0.000	0.000	0.000	8		0.000	0.000
9		0.000	0.000	1169.202	9		0.000	74.196

Figura 4.8. (a) Sumário das faltas das saídas e (b) Sumário das folgas das entradas, segundo o CCR/E/O.

SUMMARY OF OUTPUT TARGETS:				SUMMARY OF INPUT TARGETS:				
Firm	output:	1	2	3	Firm	input:	1	2
1		2.835	14.174	10630.435	1		163.000	158.039
2		5.000	50.000	50000.000	2		2000.000	233.000
3		5.000	30.000	10000.000	3		375.000	220.000
4		2.000	125.000	40000.000	4		1210.000	245.000
5		4.000	20.000	15000.000	5		230.000	223.000
6		2.342	35.133	24676.806	6		220.000	172.570
7		1.000	18.000	40000.000	7		190.000	240.000
8		3.000	50.000	35000.000	8		300.000	230.000
9		2.183	32.738	22994.297	9		205.000	160.804

Figura 4.9. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o CCR/E/O.

4.2.2 Resultados das DMUs analisadas segundo o BCC

O modelo BCC foi proposto por Banker, Charnes e Cooper, em 1984, com a premissa dos Retornos Variáveis de Escala, que são gerados através de competições imperfeitas, restrições de capital, regulamentações governamentais, entre outros.

Segundo Ferreira e Gomes (2009), as combinações possíveis entre os valores dos insumos e produtos dos ISPs são as mais variadas possíveis, o que dificulta a conclusão sobre a eficiência técnica em que elas operam e quais poderiam ser um modelo operacional que pudesse ser referência para as demais. Desta forma, algumas empresas oferecem planos com velocidades maiores que seus concorrentes; outras possuem um número maior de clientes; outras têm um *link* contratado maior que as outras; outras detêm um maior número de servidores e outras conseguem obter uma maior lucratividade. Todas estas combinações descritas só podem ser

melhores visualizadas com os cálculos dos escores de eficiência técnica dos modelos DEA, além do pressuposto das competições imperfeitas.

O BCC, tal como o CCR, possuem as mesmas formas e orientações, sendo que os resultados das eficiências técnicas, alcançadas através deste, são, geralmente, maiores que os do CCR. Assim, através da fórmula (20) do capítulo 2 é calculada a eficiência de escala que resulta em um nível de eficiência mais adequado de produção.

Os gráficos gerados pelo BCC diferem dos gerados pelo CCR, pois são não-radiais, curvilíneos e convexos, onde no CCR são lineares formando ângulos de 45°.

O *Solver* DEA proposto não calcula as folgas, faltas e alvos no modelo BCC/E/I e nem no BCC/E/O. Desta forma, foi usado o Win4DEAP para sumarizar tais resultados, apresentados nas figuras abaixo.

SUMMARY OF INPUT SLACKS:			SUMMARY OF OUTPUT SLACKS:		
firm	input:		firm	output:	
1	0.000	0.000	1	0.000	0.000
2	0.000	0.000	2	0.000	0.000
3	0.000	0.000	3	0.000	0.000
4	0.000	0.000	4	0.000	0.000
5	0.000	0.000	5	0.000	0.000
6	0.000	0.000	6	0.181	309.607
7	0.000	0.000	7	0.000	0.000
8	0.000	0.000	8	0.000	0.000
9	0.000	0.000	9	0.000	0.000

Figura 4.10. (a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/I.

SUMMARY OF INPUT TARGETS:			SUMMARY OF OUTPUT TARGETS:		
firm	input:		firm	output:	
1	163.000	215.000	1	2.000	1400.000
2	2000.000	233.000	2	5.000	50000.000
3	375.000	220.000	3	5.000	10000.000
4	1210.000	245.000	4	2.000	40000.000
5	230.000	223.000	5	4.000	15000.000
6	216.739	231.517	6	2.181	20309.607
7	190.000	240.000	7	1.000	40000.000
8	300.000	230.000	8	3.000	35000.000
9	205.000	235.000	9	2.000	20000.000

Figura 4.11. (a) Sumário dos alvos das entradas e (b) Sumário dos alvos das saídas, segundo BCC/E/I.

O BCC/E/ também foi calculado o Win4DEAP e, assim os resultados dos cálculos das folgas, faltas e alvos serão mostrados nas figuras a seguir.

SUMMARY OF INPUT SLACKS:			SUMMARY OF OUTPUT SLACKS:		
firm	input:		firm	output:	
1	0.000	0.000	1	0.000	0.000
2	0.000	0.000	2	0.000	0.000
3	0.000	0.000	3	0.000	0.000
4	0.000	0.000	4	0.000	0.000
5	0.000	0.000	5	0.000	0.000
6	0.000	0.994	6	0.000	286.624
7	0.000	0.000	7	0.000	0.000
8	0.000	0.000	8	0.000	0.000
9	0.000	0.000	9	0.000	0.000

Figura 4.12.(a) Sumário das folgas das entradas e (b) Sumário das faltas das saídas, segundo BCC/E/O.

SUMMARY OF OUTPUT TARGETS:			SUMMARY OF INPUT TARGETS:		
firm	output:		firm	input:	
1	2.000	2.000	1	163.000	215.000
2	5.000	50.000	2	2000.000	233.000
3	5.000	30.000	3	375.000	220.000
4	2.000	125.000	4	1210.000	245.000
5	4.000	20.000	5	230.000	223.000
6	2.191	32.866	6	220.000	234.006
7	1.000	18.000	7	190.000	240.000
8	3.000	50.000	8	300.000	230.000
9	2.000	30.000	9	205.000	235.000

Figura 4.13. (a) Sumário dos alvos das saídas e (b) Sumário dos alvos das entradas, segundo o BCC/E/O.

O resultado dos cálculos das eficiências técnicas dos modelos CCR e BCC foram confrontados com o intuito de serem calculadas as eficiências de escala das DMUs analisadas, além disso foram calculados os RVE, para se saber se os retornos são constantes, crescentes (ou não decrescente, i.e.; ≥ 0) ou não-crescentes (ou decrescente, i.e.; ≤ 0). Tais resultados são mostrados na tabela adiante.

De acordo com Ferreira e Gomes (2009) os padrões mais altos de eficiência técnica pura (RVE) devem estar combinados com escalas de produção mais efetivas, ou seja, devem ser capazes de gerar maiores margens de lucro. Outro objetivo da eficiência operacional das DMUs é que elas atinjam o que a DEA denomina como *Most Productive Scale Size* – MPSS, i.e., a mais eficiente escala de produção com rendimentos constantes que são custos operacionais constantes. Assim, pode ser observado que as variáveis, utilizadas como entradas e saídas das DMUs que possuem rendimentos constantes, são inalteradas, o que significa serem efetivamente eficientes e eficazes sem possuir folgas ou excessos, para atingir a posição MPSS, o que torna estas DMUs como parceiras de referência ou *benchmarks*.

As variáveis das DMUs com rendimentos decrescentes são reduzidas, assim a quantidade de insumos diminuem em uma proporção maior do que os produtos para atingir a posição MPSS. Por outro lado, quando os rendimentos são crescentes, as DMUs terão de fazer

ajustes diversos nas suas variáveis de entradas e de saídas com o intuito de atingirem a posição MPSS.

Tabela 4.2. Eficiências técnicas, de escala e RVE's das DMUs analisadas.

DMU's	Eficiências			RNC	RND	Tipo Rendimento
	CCR	BCC	Efic. de Escala	≤ 0	≥ 0	
M. Net	0,7055	1,0000	0,7055	0,7055	1,0000	CRESCENTE
Baydenet	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
ADL Link	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
NetLine	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
CePlug	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
HotLink	0,8539	0,9852	0,8667	0,8539	0,9852	DECRESCENTE
Cz.Net	0,5684	0,9842	0,5775	0,5684	1,0000	DECRESCENTE
WiNET	1,0000	1,0000	1,0000	1,0000	1,0000	CONSTANTE
Throughput	0,9164	1,0000	0,9164	0,9164	1,0000	CRESCENTE

O *Solver* DEA mostra, graficamente, as eficiências técnicas do modelo BCC em forma de colunas, que pode ser visto adiante.

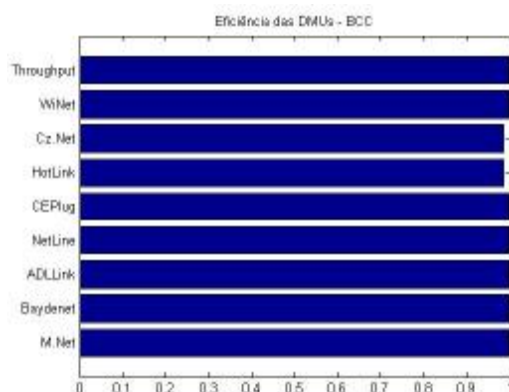


Figura 4.14. Gráfico das Eficiências Técnicas do modelo BCC gerado pelo *Solver* DEA proposto.

No DEA-Solver da Saitech foi gerado um ranqueamento das DMUs, segundo o BCC, apresentado na tabela abaixo.

Tabela 4.3. Ranqueamento das DMUs analisadas segundo BCC/E/I usando o DEA-Solver da Saitech.

Rank	DMU	Score
1	Throughput	1
1	M. Net	1
1	Baydenet	1
1	ADL Link	1
1	NetLine	1
1	CePlug	1
1	WiNET	1
8	HotLink	0,98517
9	Cz.Net	0,984202

O DEA-Solver gerou um gráfico com as eficiências técnicas do BCC que será mostrado em seguida.

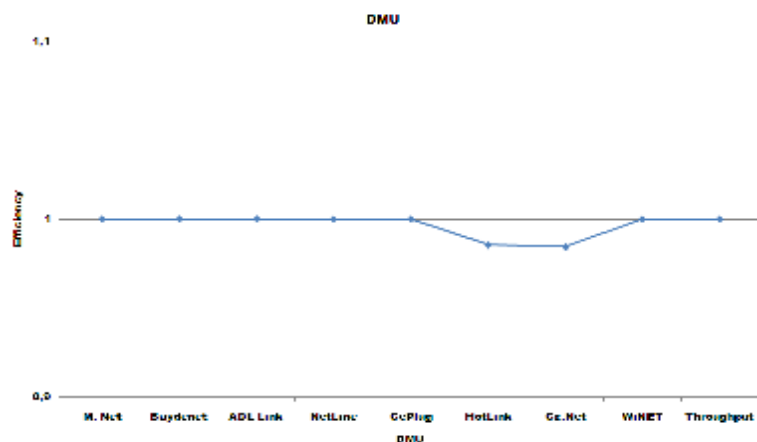


Figura 4.15. Gráfico das eficiências técnicas do modelo CCR segundo o DEA-Solver.

4.2.3 Resultados das DMUs analisadas segundo o MBF

O MBF ou Modelo Baseado em Folgas, ou Modelo Aditivo é um modelo especial da DEA que serve para se calcular, simultaneamente, a diminuição nas entradas e o aumento das saídas. Assim, o *Solver* DEA proposto analisou as DMUs, segundo o MBF, e os resultados são mostrados na figura abaixo.

===== DEB MATLAB Solver - MBF =====								
SOMA (FOLGAS + FALTAS)		FOLGAS (INPUT)			FALTAS (OUTPUT)			
DMU		I1	I2	O1	O2	O3		
M.Net	15142.23	0.00	79.88	0.00	21.18	15041.18		
Baydenet	0.00	0.00	0.00	0.00	0.00	0.00		
ADLink	0.00	0.00	0.00	0.00	0.00	0.00		
NetLine	0.00	0.00	0.00	0.00	0.00	0.00		
CePlug	0.00	0.00	0.00	0.00	0.00	0.00		
HotLink	5739.87	0.00	66.33	0.20	6.67	5666.67		
Cz.Net	18275.57	0.00	94.33	0.90	13.67	18166.67		
WiNet	0.00	0.00	0.00	0.00	0.00	0.00		
Throughput	3998.72	0.00	77.83	0.05	4.17	3916.67		

Figura 4.16. Resultados do MBF calculados com o *Solver* DEA proposto.

4.3. Análise dos resultados obtidos

A DEA serviu para comprovar que alguns ISPs oferecem sub-serviços aos usuários finais, em detrimento de seus lucros, quando, principalmente as DMUs: M.Net, HotLink, Cz.Net e Throughput mostraram os piores resultados, comparados com as outras; sendo que as DMUs CePlug e WiNet são consideradas as mais eficientes, pois possuem referências para as DMUs. Assim, estas terão de adaptar seus insumos como os destes benchmarks para se tornarem eficientes.

Por ser um método quantitativo de criação de uma fronteira de eficiência, não-paramétrico, que a DEA precisa de um grande número de DMUs para obter uma eficácia maior nos resultados obtidos. Charnes et. al. (1989) propôs uma heurística que recomenda que o número mínimo de firmas avaliadas deve ser maior ou igual a três vezes a soma das entradas e saídas, outros autores recomendam ainda mais vezes. Assim, este trabalho utilizou um número menor que o mínimo de DMUs, devido ao fato que algumas empresas de provimento de acesso à Internet nas regiões pesquisadas se negaram a responder o questionário, inviabilizando, assim o número mínimo de corporações avaliadas.

A ineficiência ou falsa eficiência deve-se ao fato de que estas têm mais usuários do que, realmente, poderiam ter para oferecer serviços com a velocidade contratada, ou seja, a quantidade de insumos é superior a quantidade necessária para que estas DMUs tornem-se eficientes. Assim, estes insumos devem ser diminuídos, mantendo a mesma proporção de saídas. Outros ISPs tornam-se cada vez mais produtivos, pois estão conectados a seus *backbones*, através de links de fibra ótica, aumentando a qualidade dos serviços oferecidos, onde as velocidades, acordadas em contrato, são quase sempre seguidas ao longo do dia.

4.4. Comentários

Este capítulo mostrou em que plataforma foi desenvolvido o *Solver* DEA proposto, e os motivos levados ao desenvolvimento no MATLAB. Foram apresentados os resultados, obtidos com o uso da DEA, nos principais modelos: CCR e BCC, com suas respectivas formas: dos Multiplicadores e de Envoltório e nas orientações de entrada e saída usando o *Solver* DEA proposto. Além de outros resultados, obtidos com o software Win4DEAP e o DEA-Solver da Saitech, tais softwares foram usados para serem calculados alvos, folgas das entradas, faltas das saídas, ranqueamento e gráficos do CCR/E/O, BCC/E/I e BCC/E/O.

Finalmente, os resultados do MBF das DMUs analisadas foram calculados com o *Solver* DEA proposto e apresentados.

Capítulo 5

Neste capítulo são feitas as considerações finais do trabalho, além da apresentação de melhorias e trabalhos futuros, acerca da pesquisa e/ou relacionada com a DEA.

5.1 Conclusões

Neste trabalho de dissertação foi utilizada uma técnica, baseada no teorema de Pareto-Koopmans e nos métodos de avaliação de produtividade de empresas, desenvolvido por Farrell. Tal técnica foi criada com o intuito de obtenção do grau de PhD de Edwardo Rhodes, em 1978, juntamente com Charnes e Cooper e foi intitulada DEA (Data Envelopment Analysis) ou Análise Envoltória de Dados.

A DEA é uma técnica de geração de fronteira de eficiência Pareto-Koopmans, i.e., onde ora se diminui a quantidade de insumos, para se produzir o mesmo produto; ou ora se aumenta os produtos, mantendo a mesma quantidade de insumos. Além do mais, a DEA consegue estimar sua fronteira de eficiência, com múltiplas entradas e múltiplas saídas, através do algoritmo Simplex de programação linear.

Este trabalho utilizou-se da DEA, para estimar várias fronteiras de eficiências, segundo os seus principais modelos: CCR e BCC, suas formas: dos multiplicadores (primal) e de envoltório (dual) e nas orientações à entrada e à saída; ou seja, o trabalho visa a tanto trabalhar com a satisfação dos clientes, com a melhoria nos serviços por eles contratados, quanto à lucratividade das empresas analisadas.

Vale ser ressaltado que os resultados das eficiências técnicas são idênticos, quando se usa o modelo CCR em qualquer forma ou orientação, bem como no BCC, mas mesmo DMUs, consideradas eficientes, podem apresentar excesso de insumos, denominadas folgas, ou falta de produtos. Logo, deve-se calcular, não só as eficiências técnicas, mas, também, os movimentos radiais para a fronteira de eficiência, as folgas das entradas, as faltas das saídas e os alvos de eficiência na fronteira. Por esta razão é dito que o cálculo é realizado em múltiplos estágios.

O *Solver* DEA proposto foi desenvolvido no MATLAB, uma plataforma de desenvolvimento matemático, baseado em matrizes, onde foi utilizada função linprog, que utiliza o algoritmo Simplex, para resolver problemas de programação linear. Destarte, este implementou todos os principais modelos, formas e orientações da DEA, para estimação da fronteira de eficiência; mas somente calculou os múltiplos estágios no CCR/E/I, pois o principal objetivo deste trabalho era aumentar o grau de satisfação do usuário do serviço. Entretanto, os múltiplos estágios, segundo o CCR/E/O, BCC/E/I e BCC/E/O, foram calculados usando o

software Win4DEAP, sendo que os gráficos de eficiência foram gerados pelo *Solver* DEA proposto, enquanto os gráficos de ranqueamento foram gerados com o DEA-Solver da SAITECH.

O *Solver* DEA proposto ainda calcula em múltiplos estágios, segundo o MBF (Modelo Baseado em Folgas) ou Modelo Aditivo, que pressupõe poder, simultaneamente, diminuir a quantidade de entradas e aumentar a quantidade de produtos produzidos.

Ao rodar todos os modelos, descobriu-se que alguns ISPs ainda trabalham, oferecendo sub-serviços aos clientes, tendo que diminuir, consideravelmente, a quantidade de usuários com o objetivo de atingir os anseios dos consumidores em relação ao contrato firmado entre as partes.

Também foi observado que o preço dos *links* vem caindo, substancialmente, no interior do Nordeste do país, pois as Teles começam a oferecer serviços de interconexão para provedores de serviços da última milha com fibra óptica. Tal benefício, junto com ferramentas de software que dividem a banda, tal como o classificador de pacotes CBQ, no Linux, fazem com que estes provedores possam chegar próximos das velocidades acordadas com os seus *backbones*, porém contratam vários serviços ADSL, tal como o Velox, de 1Mbps para proverem seus serviços de forma mais barata.

No entanto, observou-se a acurácia da técnica DEA que será utilizada pelo autor deste trabalho, para outros fins acadêmicos e que serão apresentados na seção seguinte.

5.2 Trabalhos futuros

Como extensão deste e outros trabalhos futuros são propostos:

- (i) Implementação dos cálculos, em múltiplos estágios da DEA, no CCR/E/O, BCC/E/I e BCC/E/O;
- (ii) Parametrização do *Solver* DEA proposto para que usuários possam usá-lo mais facilmente;
- (iii) Instigar o desempenho do *Solver* DEA proposto com um maior número de DMUs para avaliar seu desempenho em relação a outros softwares DEA;
- (iv) Criação de gráficos de fronteira eficiente no CCR e BCC;
- (v) Implementação de um Web DEA livre e gratuito, para calcular os principais modelos da DEA em múltiplos estágios, com os respectivos gráficos, provendo uma maneira ubíqua de

se estimar fronteiras de eficiência entre empresas, setores de empresas, órgãos governamentais, entre outros;

(vi) Utilização do Modelo de Supereficiência da DEA, no escalonamento de processos de Sistemas Operacionais e/ou em grades computacionais, além de testar a DEA em algoritmos de roteamento em redes *ad-hoc* sem fio.

Referências Bibliográficas

- BARFORD, P.; SOMMERS, J. (2004). Comparing probe-based and router-based packet-loss measurement. *IEEE Internet Computing*, 8(5):50–56.
- CHARNES, A.; COOPER, W.W. & RHODES, E. (1978). Measuring efficiency of decision making units, *European Journal of Operational Research*, 3, pp. 429-444.
- CHARNES, A.; COOPER W.W.; GOLANY, B.; SEIFORD, L.M.; AND STUTZ, J. (1985). “Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions.” *Journal of Econometrics* 30, 91-107.
- CHARNES, A.; COOPER, W. W.; AHN, TAE SIK; ARNOLD, V. (1989). "DEA and Ratio Efficiency Analyses for Public Institutions of Higher Learning in Texas". *Research in Governmental and Nonprofit Accounting* (1989), 5:165-185.
- CHAVES, A. C. A; THOMAZ, A. C. F. (2008). “Gestão Pública e Pesquisa Operacional: avaliação de desempenho em Agências da Previdência Social”, *Revista do Serviço Público*, Brasília 59 (2): 221-236 Abr/Jun 2008.
- CHERCHYE, L.; T. KUOSMANEN and T. POST. (2000). "What is the Economic Meaning of FDH?", *Journal of Productivity Analysis* 13 (3), 263-267.
- COELLI, T. (1996). A guide to DEAP version 2.1: a data envelopment analysis (computer) program. Center for Efficiency and Productivity Analysis. Department of Econometrics, New England University.
- COOPER, W.W.; SEIFORD, L.M. & ZHU, J. (2004). *Data Envelopment Analysis: Models and interpretations*. In *Handbook on Data Envelopment Analysis*, Boston: Kluwer Academic Publisher.
- COOPER, W.W.; SEIFORD, L.M. & TONE, K. (2007). *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Kluwer Academic Publishers, USA.
- DANTZIG, G. (1963). *Linear Programming and Extensions*, Princeton University Press, Princeton.
- ENZO, B. MARIANO; ALMEIDA, R. MARIANA; REBELATO, A. N. DAISY. (2006). “Peculiaridades da Análise Envoltória de Dados”, XII SIMPEP, 2006.

- FERREIRA, CARLOS M. DE CARVALHO; GOMES, PROVEZANO. (2009). "Introdução à Análise Envoltória de Dados". Editora UFV, 1ª edição.
- KARMAKAR, N. (1984). "A New Polynomial Time Algorithm for Linear Programming", *Combinatoria* 4, pp.373-395.
- KOOPMANS, TJALLING CHARLES. (1951). *Activity analysis of production and allocation : proceedings of a conference*, John Wiley and Sons, New York.
- LEVERTY, J.T.; GRACE, M.F. (2007). *Issues in Measuring the Efficiency of Property-Liability Insurers*, Palgrave Macmillan.
- LINS, M. P. E. ; CALÔBA, G.M. (2006). *Programação Linear com aplicações em teoria dos jogos e avaliação de desempenho (Data Envelopment Analysis)*. 1. ed. Rio de Janeiro: Interciência, v. 1. 299 p.
- MACEDO, M. A. S.; SANTOS, Ruthberg dos ; ANDRADE, Jarbas Emanuel Antunes; BENAC, Marcos Azevedo. (2005). *Avaliação da Eficiência dos Investimentos em TI de Empresas de Tecnologia através da Análise Envoltória de Dados (DEA)*. In: VIII SEMEAD - Seminários em Administração da USP, São Paulo, SP. Anais do VIII SEMEAD-USP, 2005.
- NEUMANN, J VON. (1945). A model of general economic equilibrium. *Review of Economic Studies*, XIII,1–9.
- QUINTANILHA, J. A.; HO, L. L. (2006). A performance index developed by data envelopment analysis (DEA) to compare the efficiency of fire risk monitoring actions in municipalities of Brazilian Amazon region. In: *Accuracy 2006: The Seventh International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, 2006, Lisboa, Portugal. The Seventh International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences. Lisboa - Portugal : Instituto Geográfico Português, p. 624-632.
- REINALDO, R. R. P.; POSSAMAI, O.; THOMAZ, A. C. F. (2002). *Avaliando a eficiência em unidades de ensino fundamental de Fortaleza usando técnicas de análise envoltória de dados (DEA)*. *Revista Científica da Faculdade Lourenço Filho*, [S.l.], v. 2, n. 1. Disponível em: <<http://www.flf.edu.br/midias/FLF.EDU/21.pdf>>. Acesso em 30 mar. 2010.
- RESENDE NETO, ANTONIO DE LARA. (2006). *Uma Proposta de Construção de Indicador de Performance de Fundos de Investimento*, UnB, 86 p.
- RAJ JAIN. (1991). "The Art of Computer Systems. Performance Analysis", Wiley & Sons.

VILLARROEL DÁVALOS, R. D.; MORALES, A. T. ; MAGAJEWSKI, F.; SCHLICKMANN, T. (1999) . Uma proposta para modelar os indicadores da saúde através de uma análise envoltória de dados. In: XXXI SBPO - Simpósio Brasileiro de Pesquisa Operacional, Juíz de Fora. XXXI SBPO.

ZIVIANI, ARTUR; DUARTE, O. C. M. B. (2005). Mini-curso do XXIII Simpósio Brasileiro de Redes de Computadores SBRC'2005. Sociedade Brasileira de Computação, 2005.

ANEXOS

% DEA - CCR - Envelopment Mode - Input Oriented – MATLAB Code
 % Developed by Francisco Daladier Marques Junior
 % contact: daladierjr@ifpb.edu.br

```
clc;
clear;
```

```
f= [1;0;0;0;0;0;0;0;0;]; %objective function
```

```
b_A = [ 0; 0; 2;2;1400]; %constraints
```

```
DMU_A = [ 163 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          215 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_B = [ 0;0; 5;50;50000]; %constraints
```

```
DMU_B = [ 2000 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          233 -215 -233 -220 -245 -223 -235 -240 -230 -235 ; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_C = [ 0;0; 5;30;10000]; %constraints
```

```
DMU_C = [ 375 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          220 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_D = [ 0;0; 2;125;40000]; %constraints
```

```
DMU_D = [ 1210 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          245 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_E = [ 0;0; 4;20;15000]; %constraints
```

```
DMU_E = [ 230 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          223 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_F = [0;0; 2;30;20000]; %constraints
```

```
DMU_F = [ 220 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
          235 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
          0 2 5 5 2 4 2 1 3 2; % Output_1
          0 2 50 30 125 20 30 18 50 30; % Output_2
          0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3
```

```
b_G = [ 0;0; 1;18;4000]; %constraints
```

```
DMU_G = [ 190 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
```

```

240 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

b_H = [ 0;0;3;50;35000;]; %constraints

DMU_H = [ 300 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
230 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

b_I = [ 0;0; 2;30;20000]; %constraints

DMU_I = [205 -163 -2000 -375 -1210 -230 -220 -190 -300 -205; % Input_1
235 -215 -233 -220 -245 -223 -235 -240 -230 -235; % Input_2
0 2 5 5 2 4 2 1 3 2; % Output_1
0 2 50 30 125 20 30 18 50 30; % Output_2
0 1400 50000 10000 40000 15000 20000 4000 35000 20000;]; % Output_3

lb=zeros(10,1);

% Calculation of Technical Efficiency

[lambda_A,FVAL_A,exitflag,output,lambda]=linprog(f,-DMU_A,-b_A,[],[],lb);

[lambda_B,FVAL_B,exitflag,output,lambda]=linprog(f,-DMU_B,-b_B,[],[],lb);

[lambda_C,FVAL_C,exitflag,output,lambda]=linprog(f,-DMU_C,-b_C,[],[],lb);

[lambda_D,FVAL_D,exitflag,output,lambda]=linprog(f,-DMU_D,-b_D,[],[],lb);

[lambda_E,FVAL_E,exitflag,output,lambda]=linprog(f,-DMU_E,-b_E,[],[],lb);

[lambda_F,FVAL_F,exitflag,output,lambda]=linprog(f,-DMU_F,-b_F,[],[],lb);

[lambda_G,FVAL_G,exitflag,output,lambda]=linprog(f,-DMU_G,-b_G,[],[],lb);

[lambda_H,FVAL_H,exitflag,output,lambda]=linprog(f,-DMU_H,-b_H,[],[],lb);

[lambda_I,FVAL_I,exitflag,output,lambda]=linprog(f,-DMU_I,-b_I,[],[],lb);

for i=2:10
    if lambda_A(i,1)>0.01
        disp('O valor DMU_A para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_A(i,1));
        lambdaA(1,i)=lambda_A(i,1);
    end
end

LambdaA_index=find(lambdaA); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaA_index)
    LambdaA_index(1,j)=LambdaA_index(1,j)-1;
end

```

```

end
lambdaA=lambdaA(lambdaA~=0);

for i=2:10
    if lambda_B(i,1)>0.01
        disp('O valor DMU_B para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_B(i,1));
        lambdaB(1,i)=lambda_B(i,1);

    end
end

LambdaB_index=find(lambdaB); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaB_index)
    LambdaB_index(1,j)=LambdaB_index(1,j)-1;
end
lambdaB=lambdaB(lambdaB~=0);

for i=2:10
    if lambda_C(i,1)>0.01
        disp('O valor DMU_C para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_C(i,1));
        lambdaC(1,i)=lambda_C(i,1);

    end
end

LambdaC_index=find(lambdaC); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaC_index)
    LambdaC_index(1,j)=LambdaC_index(1,j)-1;
end
lambdaC=lambdaC(lambdaC~=0);

for i=2:10
    if lambda_D(i,1)>0.01
        disp('O valor DMU_D para o Lambda ');
        disp(i-1);
        disp('é');
        disp(lambda_D(i,1));
        lambdaD(1,i)=lambda_D(i,1);

    end
end

LambdaD_index=find(lambdaD); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaD_index)
    LambdaD_index(1,j)=LambdaD_index(1,j)-1;
end
lambdaD=lambdaD(lambdaD~=0);

for i=2:10
    if lambda_E(i,1)>0.01

```

```

disp('O valor DMU_E para o Lambda ');
disp(i-1);
disp('é:');
disp(lambda_E(i,1));
lambdaE(1,i)=lambda_E(i,1);

end
end

LambdaE_index=find(lambdaE); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaE_index)
    LambdaE_index(1,j)=LambdaE_index(1,j)-1;
end
lambdaE=lambdaE(lambdaE~=0);

for i=2:10
    if lambda_F(i,1)>0.01
        disp('O valor DMU_F para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_F(i,1));
        lambdaF(1,i)=lambda_F(i,1);

    end
end

LambdaF_index=find(lambdaF); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaF_index)
    LambdaF_index(1,j)=LambdaF_index(1,j)-1;
end
lambdaF=lambdaF(lambdaF~=0);

for i=2:10
    if lambda_G(i,1)>0.01
        disp('O valor DMU_G para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_G(i,1));
        lambdaG(1,i)=lambda_G(i,1);

    end
end

LambdaG_index=find(lambdaG); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaG_index)
    LambdaG_index(1,j)=LambdaG_index(1,j)-1;
end
lambdaG=lambdaG(lambdaG~=0);

for i=2:10
    if lambda_H(i,1)>0.01
        disp('O valor DMU_H para o Lambda ');
        disp(i-1);
    end
end

```

```

        disp('é:');
        disp(lambda_H(i,1));
        lambdaH(1,i)=lambda_H(i,1);

    end
end

LambdaH_index=find(lambdaH); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaH_index)
    LambdaH_index(1,j)=LambdaH_index(1,j)-1;
end
lambdaH=lambdaH(lambdaH~=0);

for i=2:10
    if lambda_I(i,1)>0.01
        disp('O valor DMU_I para o Lambda ');
        disp(i-1);
        disp('é:');
        disp(lambda_I(i,1));
        lambdaI(1,i)=lambda_I(i,1);

    end
end

LambdaI_index=find(lambdaI); % find indexes from nonzeros values, i.e., find peers from actual DMU
for j=1:length(LambdaI_index)
    LambdaI_index(1,j)=LambdaI_index(1,j)-1;
end
lambdaI=lambdaI(lambdaI~=0);

% Calculation of Slacks

SomaA1=0;
SomaA2=0;
SomaA3=0;
SomaA4=0;
SomaA5=0;
for i=1:length(LambdaA_index)
    SomaA1 = SomaA1 + (lambdaA(1,i)) * (-DMU_A(1,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA2 = SomaA2 + (lambdaA(1,i)) * (-DMU_A(2,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA3 = SomaA3 + (lambdaA(1,i)) * (-DMU_A(3,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA4 = SomaA4 + (lambdaA(1,i)) * (-DMU_A(4,LambdaA_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaA5 = SomaA5 + (lambdaA(1,i)) * (-DMU_A(5,LambdaA_index(1,i)+1));
end

inputSlack1_A = (FVAL_A * DMU_A(1,1)) - (SomaA1);

```

```

inputSlack2_A = (FVAL_A * DMU_A(2,1)) - (SomaA2);
outputSlack1_A = (FVAL_A * DMU_A(3,1)) - (SomaA3);
outputSlack2_A = (FVAL_A * DMU_A(4,1)) - (SomaA4);
outputSlack3_A = (FVAL_A * DMU_A(5,1)) - (SomaA5);

```

```

SomaB1=0;
SomaB2=0;
SomaB3=0;
SomaB4=0;
SomaB5=0;

```

```

for i=1:length(LambdaB_index)
    SomaB1 = SomaB1 + (lambdaB(1,i)) * (-DMU_B(1,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB2 = SomaB2 + (lambdaB(1,i)) * (-DMU_B(2,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB3 = SomaB3 + (lambdaB(1,i)) * (-DMU_B(3,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB4 = SomaB4 + (lambdaB(1,i)) * (-DMU_B(4,LambdaB_index(1,i)+1));
end
for i=1:length(LambdaB_index)
    SomaB5 = SomaB5 + (lambdaB(1,i)) * (-DMU_B(5,LambdaB_index(1,i)+1));
end

```

```

inputSlack1_B = (FVAL_B * DMU_B(1,1)) - (SomaB1);
inputSlack2_B = (FVAL_B * DMU_B(2,1)) - (SomaB2);
outputSlack1_B = (FVAL_B * DMU_B(3,1)) - (SomaB3);
outputSlack2_B = (FVAL_B * DMU_B(4,1)) - (SomaB4);
outputSlack3_B = (FVAL_B * DMU_B(5,1)) - (SomaB5);

```

```

SomaC1=0;
SomaC2=0;
SomaC3=0;
SomaC4=0;
SomaC5=0;

```

```

for i=1:length(LambdaC_index)
    SomaC1 = SomaC1 + (lambdaC(1,i)) * (-DMU_C(1,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC2 = SomaC2 + (lambdaC(1,i)) * (-DMU_C(2,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC3 = SomaC3 + (lambdaC(1,i)) * (-DMU_C(3,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaC_index)
    SomaC4 = SomaC4 + (lambdaC(1,i)) * (-DMU_C(4,LambdaC_index(1,i)+1));
end
for i=1:length(LambdaA_index)
    SomaC5 = SomaC5 + (lambdaC(1,i)) * (-DMU_C(5,LambdaC_index(1,i)+1));
end

```

```

inputSlack1_C = (FVAL_C * DMU_C(1,1)) - (SomaC1);
inputSlack2_C = (FVAL_C * DMU_C(2,1)) - (SomaC2);
outputSlack1_C = (FVAL_C * DMU_C(3,1)) - (SomaC3);

```

```
outputSlack2_C = (FVAL_C * DMU_C(4,1)) - (SomaC4);
outputSlack3_C = (FVAL_C * DMU_C(5,1)) - (SomaC5);
```

```
SomaD1=0;
SomaD2=0;
SomaD3=0;
SomaD4=0;
SomaD5=0;
```

```
for i=1:length(LambdaD_index)
    SomaD1 = SomaD1 + (lambdaD(1,i)) * (-DMU_D(1,LambdaD_index(1,i)+1));
    single(SomaD1);
end
for i=1:length(LambdaD_index)
    SomaD2 = SomaD2 + (lambdaD(1,i)) * (-DMU_D(2,LambdaD_index(1,i)+1));
    single(SomaD2);
end
for i=1:length(LambdaD_index)
    SomaD3 = SomaD3 + (lambdaD(1,i)) * (-DMU_D(3,LambdaD_index(1,i)+1));
end
for i=1:length(LambdaD_index)
    SomaD4 = SomaD4 + (lambdaD(1,i)) * (-DMU_D(4,LambdaD_index(1,i)+1));
end
for i=1:length(LambdaD_index)
    SomaD5 = SomaD5 + (lambdaD(1,i)) * (-DMU_D(5,LambdaD_index(1,i)+1));
end
```

```
inputSlack1_D = (FVAL_D * DMU_D(1,1)) - (SomaD1);
inputSlack2_D = (FVAL_D * DMU_D(2,1)) - (SomaD2);
outputSlack1_D = (FVAL_D * DMU_D(3,1)) - (SomaD3);
outputSlack2_D = (FVAL_D * DMU_D(4,1)) - (SomaD4);
outputSlack3_D = (FVAL_D * DMU_D(5,1)) - (SomaD5);
```

```
SomaE1=0;
SomaE2=0;
SomaE3=0;
SomaE4=0;
SomaE5=0;
```

```
for i=1:length(LambdaE_index)
    SomaE1 = SomaE1 + (lambdaE(1,i)) * (-DMU_E(1,LambdaE_index(1,i)+1));
    single(SomaE1);
end
for i=1:length(LambdaE_index)
    SomaE2 = SomaE2 + (lambdaE(1,i)) * (-DMU_E(2,LambdaE_index(1,i)+1));
    single(SomaE2);
end
for i=1:length(LambdaE_index)
    SomaE3 = SomaE3 + (lambdaE(1,i)) * (-DMU_E(3,LambdaE_index(1,i)+1));
end
for i=1:length(LambdaE_index)
    SomaE4 = SomaE4 + (lambdaE(1,i)) * (-DMU_E(4,LambdaE_index(1,i)+1));
end
for i=1:length(LambdaE_index)
    SomaE5 = SomaE5 + (lambdaE(1,i)) * (-DMU_E(5,LambdaE_index(1,i)+1));
end
```

```
inputSlack1_E = (FVAL_E * DMU_E(1,1)) - (SomaE1);
inputSlack2_E = (FVAL_E * DMU_E(2,1)) - (SomaE2);
```



```

outputSlack1_E = (FVAL_E * DMU_E(3,1)) - (SomaE3);
outputSlack2_E = (FVAL_E * DMU_E(4,1)) - (SomaE4);
outputSlack3_E = (FVAL_E * DMU_E(5,1)) - (SomaE5);

```

```

SomaF1=0;
SomaF2=0;
SomaF3=0;
SomaF4=0;
SomaF5=0;

```

```

for i=1:length(LambdaF_index)
    SomaF1 = SomaF1 + (lambdaF(1,i)) * (-DMU_F(1,LambdaF_index(1,i)+1));
    single(SomaF1);
end
for i=1:length(LambdaF_index)
    SomaF2 = SomaF2 + (lambdaF(1,i)) * (-DMU_F(2,LambdaF_index(1,i)+1));
    single(SomaF2);
end
for i=1:length(LambdaF_index)
    SomaF3 = SomaF3 + (lambdaF(1,i)) * (-DMU_F(3,LambdaF_index(1,i)+1));
end
for i=1:length(LambdaF_index)
    SomaF4 = SomaF4 + (lambdaF(1,i)) * (-DMU_F(4,LambdaF_index(1,i)+1));
end
for i=1:length(LambdaF_index)
    SomaF5 = SomaF5 + (lambdaF(1,i)) * (-DMU_F(5,LambdaF_index(1,i)+1));
end

```

```

inputSlack1_F = (FVAL_F * DMU_F(1,1))- (SomaF1);
inputSlack2_F = (FVAL_F * DMU_F(2,1)) - (SomaF2);
outputSlack1_F = (FVAL_F * DMU_F(3,1)) - (SomaF3);
outputSlack2_F = (FVAL_F * DMU_F(4,1)) - (SomaF4);
outputSlack3_F = (FVAL_F * DMU_F(5,1)) - (SomaF5);

```

```

SomaG1=0;
SomaG2=0;
SomaG3=0;
SomaG4=0;
SomaG5=0;

```

```

for i=1:length(LambdaG_index)
    SomaG1 = SomaG1 + (lambdaG(1,i)) * (-DMU_G(1,LambdaG_index(1,i)+1));
    single(SomaF1);
end
for i=1:length(LambdaG_index)
    SomaG2 = SomaG2 + (lambdaG(1,i)) * (-DMU_G(2,LambdaG_index(1,i)+1));
    single(SomaF2);
end
for i=1:length(LambdaG_index)
    SomaG3 = SomaG3 + (lambdaG(1,i)) * (-DMU_G(3,LambdaG_index(1,i)+1));
end
for i=1:length(LambdaG_index)
    SomaG4 = SomaG4 + (lambdaG(1,i)) * (-DMU_G(4,LambdaG_index(1,i)+1));
end
for i=1:length(LambdaG_index)
    SomaG5 = SomaG5 + (lambdaG(1,i)) * (-DMU_G(5,LambdaG_index(1,i)+1));
end

```

```

inputSlack1_G = (FVAL_G * DMU_G(1,1))- (SomaG1);

```

```

inputSlack2_G = (FVAL_G * DMU_G(2,1)) - (SomaG2);
outputSlack1_G = (FVAL_G * DMU_G(3,1)) - (SomaG3);
outputSlack2_G = (FVAL_G * DMU_G(4,1)) - (SomaG4);
outputSlack3_G = (FVAL_G * DMU_G(5,1)) - (SomaG5);

```

```

SomaH1=0;
SomaH2=0;
SomaH3=0;
SomaH4=0;
SomaH5=0;

```

```

for i=1:length(LambdaH_index)
    SomaH1 = SomaH1 + (lambdaH(1,i)) * (-DMU_H(1,LambdaH_index(1,i)+1));
    single(SomaH1);
end
for i=1:length(LambdaH_index)
    SomaH2 = SomaH2 + (lambdaH(1,i)) * (-DMU_H(2,LambdaH_index(1,i)+1));
    single(SomaH2);
end
for i=1:length(LambdaH_index)
    SomaH3 = SomaH3 + (lambdaH(1,i)) * (-DMU_H(3,LambdaH_index(1,i)+1));
end
for i=1:length(LambdaH_index)
    SomaH4 = SomaH4 + (lambdaH(1,i)) * (-DMU_H(4,LambdaH_index(1,i)+1));
end
for i=1:length(LambdaH_index)
    SomaH5 = SomaH5 + (lambdaH(1,i)) * (-DMU_H(5,LambdaH_index(1,i)+1));
end

```

```

inputSlack1_H = (FVAL_H * DMU_H(1,1)) - (SomaH1);
inputSlack2_H = (FVAL_H * DMU_H(2,1)) - (SomaH2);
outputSlack1_H = (FVAL_H * DMU_H(3,1)) - (SomaH3);
outputSlack2_H = (FVAL_H * DMU_H(4,1)) - (SomaH4);
outputSlack3_H = (FVAL_H * DMU_H(5,1)) - (SomaH5);

```

```

SomaI1=0;
SomaI2=0;
SomaI3=0;
SomaI4=0;
SomaI5=0;

```

```

for i=1:length(LambdaI_index)
    SomaI1 = SomaI1 + (lambdaI(1,i)) * (-DMU_I(1,LambdaI_index(1,i)+1));
    single(SomaI1);
end
for i=1:length(LambdaI_index)
    SomaI2 = SomaI2 + (lambdaI(1,i)) * (-DMU_I(2,LambdaI_index(1,i)+1));
    single(SomaI2);
end
for i=1:length(LambdaI_index)
    SomaI3 = SomaI3 + (lambdaI(1,i)) * (-DMU_I(3,LambdaI_index(1,i)+1));
end
for i=1:length(LambdaI_index)
    SomaI4 = SomaI4 + (lambdaI(1,i)) * (-DMU_I(4,LambdaI_index(1,i)+1));
end
for i=1:length(LambdaI_index)
    SomaI5 = SomaI5 + (lambdaI(1,i)) * (-DMU_I(5,LambdaI_index(1,i)+1));
end

```

```

inputSlack1_I = (FVAL_I * DMU_I(1,1)) - (SomaI1);
inputSlack2_I = (FVAL_I * DMU_I(2,1)) - (SomaI2);
outputSlack1_I = (FVAL_I * DMU_I(3,1)) - (SomaI3);
outputSlack2_I = (FVAL_I * DMU_I(4,1)) - (SomaI4);
outputSlack3_I = (FVAL_I * DMU_I(5,1)) - (SomaI5);

```

```

[lambda_A,FVAL_A,exitflag,output,lambda]=linprog(f,-DMU_A,-b_A,[],[],lb);

[lambda_B,FVAL_B,exitflag,output,lambda]=linprog(f,-DMU_B,-b_B,[],[],lb);

[lambda_C,FVAL_C,exitflag,output,lambda]=linprog(f,-DMU_C,-b_C,[],[],lb);

[lambda_D,FVAL_D,exitflag,output,lambda]=linprog(f,-DMU_D,-b_D,[],[],lb);

[lambda_E,FVAL_E,exitflag,output,lambda]=linprog(f,-DMU_E,-b_E,[],[],lb);

[lambda_F,FVAL_F,exitflag,output,lambda]=linprog(f,-DMU_F,-b_F,[],[],lb);

[lambda_G,FVAL_G,exitflag,output,lambda]=linprog(f,-DMU_G,-b_G,[],[],lb);

[lambda_H,FVAL_H,exitflag,output,lambda]=linprog(f,-DMU_H,-b_H,[],[],lb);

[lambda_I,FVAL_I,exitflag,output,lambda]=linprog(f,-DMU_I,-b_I,[],[],lb);

```

% Targets of the inputs from DMU

```

if (FVAL_A~=1)
    Radial1_A=FVAL_A*DMU_A(1,1);
    Radial2_A=FVAL_A*DMU_A(2,1);
    Radial3_A=FVAL_A*DMU_A(3,2);
    Radial4_A=FVAL_A*DMU_A(4,2);
    Radial5_A=FVAL_A*DMU_A(5,2);
end

if (FVAL_B~=1)
    Radial1_B=FVAL_B*DMU_B(1,1);
    Radial2_B=FVAL_B*DMU_B(2,1);
    Radial3_B=FVAL_B*DMU_B(3,3);
    Radial4_B=FVAL_B*DMU_B(4,3);
    Radial5_B=FVAL_B*DMU_B(5,3);
end

if (FVAL_C~=1)
    Radial1_C=FVAL_C*DMU_C(1,1);
    Radial2_C=FVAL_C*DMU_C(2,1);
    Radial3_C=FVAL_C*DMU_C(3,4);
    Radial4_C=FVAL_C*DMU_C(4,4);
    Radial5_C=FVAL_C*DMU_C(5,4);
end

if (FVAL_D~=1)
    Radial1_D=FVAL_D*DMU_D(1,1);
    Radial2_D=FVAL_D*DMU_D(2,1);

```

```

Radial3_D=FVAL_D*DMU_D(3,5);
Radial4_D=FVAL_D*DMU_D(4,5);
Radial5_D=FVAL_D*DMU_D(5,5);

end

if (FVAL_E~=1)
    Radial1_E=FVAL_E*DMU_E(1,1);
    Radial2_E=FVAL_E*DMU_E(2,1);
    Radial3_E=FVAL_E*DMU_E(3,6);
    Radial4_E=FVAL_E*DMU_E(4,6);
    Radial5_E=FVAL_E*DMU_E(5,6);

end

if (FVAL_F~=1)
    Radial1_F=FVAL_F*DMU_F(1,1);
    Radial2_F=FVAL_F*DMU_F(2,1);
    Radial3_F=FVAL_F*DMU_F(3,7);
    Radial4_F=FVAL_F*DMU_F(4,7);
    Radial5_F=FVAL_F*DMU_F(5,7);

end

if (FVAL_G~=1)
    Radial1_G=FVAL_G*DMU_G(1,1);
    Radial2_G=FVAL_G*DMU_G(2,1);
    Radial3_G=FVAL_G*DMU_G(3,8);
    Radial4_G=FVAL_G*DMU_G(4,8);
    Radial5_G=FVAL_G*DMU_G(5,8);

end

if (FVAL_H~=1)
    Radial1_H=FVAL_H*DMU_H(1,1);
    Radial2_H=FVAL_H*DMU_H(2,1);
    Radial3_H=FVAL_H*DMU_A(3,9);
    Radial4_H=FVAL_H*DMU_H(4,9);
    Radial5_H=FVAL_H*DMU_H(5,9);

end

if (FVAL_I~=1)
    Radial1_I=FVAL_I*DMU_I(1,1);
    Radial2_I=FVAL_I*DMU_I(2,1);
    Radial3_I=FVAL_I*DMU_I(3,10);
    Radial4_I=FVAL_I*DMU_A(4,10);
    Radial5_I=FVAL_I*DMU_A(5,10);

end

Target1_A = Radial1_A - inputSlack1_A;
Target2_A = Radial2_A - inputSlack2_A;
Target3_A = outputSlack1_A - DMU_A(3,2);
Target4_A = outputSlack2_A - DMU_A(4,2);
Target5_A = outputSlack3_A - DMU_A(5,2);

Target1_B = Radial1_B - inputSlack1_B;
Target2_B = Radial2_B - inputSlack2_B;

```

```

Target3_B = outputSlack1_B - DMU_A(3,3);
Target4_B = outputSlack2_B - DMU_A(4,3);
Target5_B = outputSlack3_B - DMU_A(5,3);

```

```

Target1_C = Radial1_C - inputSlack1_C;
Target2_C = Radial2_C - inputSlack2_C;
Target3_C = outputSlack1_C - DMU_A(3,4);
Target4_C = outputSlack2_C - DMU_A(4,4);
Target5_C = outputSlack3_C - DMU_A(5,4);

```

```

Target1_D = Radial1_D - inputSlack1_D;
Target2_D = Radial2_D - inputSlack2_D;
Target3_D = outputSlack1_D - DMU_A(3,5);
Target4_D = outputSlack2_D - DMU_A(4,5);
Target5_D = outputSlack3_D - DMU_A(5,5);

```

```

Target1_E = Radial1_E - inputSlack1_E;
Target2_E = Radial2_E - inputSlack2_E;
Target3_E = outputSlack1_E - DMU_A(3,6);
Target4_E = outputSlack2_E - DMU_A(4,6);
Target5_E = outputSlack3_E - DMU_A(5,6);

```

```

Target1_F = Radial1_F - inputSlack1_F;
Target2_F = Radial2_F - inputSlack2_F;
Target3_F = outputSlack1_F - DMU_A(3,7);
Target4_F = outputSlack2_F - DMU_A(4,7);
Target5_F = outputSlack3_F - DMU_A(5,7);

```

```

Target1_G = Radial1_G - inputSlack1_G;
Target2_G = Radial2_G - inputSlack2_G;
Target3_G = outputSlack1_G - DMU_A(3,8);
Target4_G = outputSlack2_G - DMU_A(4,8);
Target5_G = outputSlack3_G - DMU_A(5,8);

```

```

Target1_H = Radial1_H - inputSlack1_H;
Target2_H = Radial2_H - inputSlack2_H;
Target3_H = outputSlack1_H - DMU_A(3,9);
Target4_H = outputSlack2_H - DMU_A(4,9);
Target5_H = outputSlack3_H - DMU_A(5,9);

```

```

Target1_I = Radial1_I - inputSlack1_I;
Target2_I = Radial2_I - inputSlack2_I;
Target3_I = outputSlack1_I - DMU_A(3,10);
Target4_I = outputSlack2_I - DMU_A(4,10);
Target5_I = outputSlack3_I - DMU_A(5,10);

```

```

clc;

```

```

% Additive Model to Calculate Slacks

```

```

%DMU_Eq = [ 2 4 3 2 1 3 -1 0 0;
%          6 8 12 6 6 15 0 1 0;
%          14 16 12 4 3 3 0 0 1;
%          1 1 1 1 1 1 0 0 0;];
%A B C D E F G
DMU_Eq = [ 2 5 5 2 4 2 1 3 2 -1 0 0 0 0; % Output_1

```

```

2 50 30 125 20 30 18 50 30 0 -1 0 0 0; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 0 0 -1 0 0; % Output_3
163 2000 375 1210 230 220 190 300 205 0 0 0 1 0; % Input_1
215 233 220 245 223 235 240 230 235 0 0 0 1; % Input_2

```

```

objective = [ 0;0;0;0;0;0;0;0;1;1;1;1;1];

```

```

A_Eq = DMU_Eq(:,1);
B_Eq = DMU_Eq(:,2);
C_Eq = DMU_Eq(:,3);
D_Eq = DMU_Eq(:,4);
E_Eq = DMU_Eq(:,5);
F_Eq = DMU_Eq(:,6);
G_Eq = DMU_Eq(:,7);
H_Eq = DMU_Eq(:,8);
I_Eq = DMU_Eq(:,9);

```

```

lb2 = zeros(14,1);

```

```

[XA,SLACKA,EXITFLAG_SA,OUTPUT_A,lambdaXA]=linprog(-objective,[],[],DMU_Eq,A_Eq,lb2);
[XB,SLACKB,EXITFLAG_SB,OUTPUT_B,lambdaXB]=linprog(-objective,[],[],DMU_Eq,B_Eq,lb2);
[XC,SLACKC,EXITFLAG_SC,OUTPUT_C,lambdaXC]=linprog(-objective,[],[],DMU_Eq,C_Eq,lb2);
[XD,SLACKD,EXITFLAG_SD,OUTPUT_D,lambdaXD]=linprog(-objective,[],[],DMU_Eq,D_Eq,lb2);
[XE,SLACKE,EXITFLAG_SE,OUTPUT_E,lambdaXE]=linprog(-objective,[],[],DMU_Eq,E_Eq,lb2);
[XF,SLACKF,EXITFLAG_SF,OUTPUT_F,lambdaXF]=linprog(-objective,[],[],DMU_Eq,F_Eq,lb2);
options = optimset('LargeScale','off','Simplex','on','TolFun',1e-8,'MaxIter',10);
[XG,SLACKG,EXITFLAG_SG,OUTPUT_G,lambdaXG]=linprog(-
objective,[],[],DMU_Eq,G_Eq,lb2,[],[],options);
[XH,SLACKH,EXITFLAG_SH,OUTPUT_H,lambdaXH]=linprog(-
objective,[],[],DMU_Eq,H_Eq,lb2,[],[],options);
[XI,SLACKI,EXITFLAG_SI,OUTPUT_I,lambdaXI]=linprog(-
objective,[],[],DMU_Eq,I_Eq,lb2,[],[],options);

```

```

for a=1:14
    SL_A(a)=XA(a);
end

```

```

Slack1_A = SL_A(1,10);
Slack2_A = SL_A(1,11);
Slack3_A = SL_A(1,12);
Excess1_A = SL_A(1,13);
Excess2_A = SL_A(1,14);

```

```

for b=1:14
    SL_B(b)=XB(b);
end

```

```

Slack1_B = SL_B(1,10);
Slack2_B = SL_B(1,11);
Slack3_B = SL_B(1,12);
Excess1_B = SL_B(1,13);
Excess2_B = SL_B(1,14);

```

```

for c=1:14
    SL_C(c)=XC(c);
end

```

```

Slack1_C = SL_C(1,10);

```

```

    Slack2_C = SL_C(1,11);
    Slack3_C = SL_C(1,12);
    Excess1_C = SL_C(1,13);
    Excess2_C = SL_C(1,14);

for d=1:14
    SL_D(d)=XD(d);
end

    Slack1_D = SL_D(1,10);
    Slack2_D = SL_D(1,11);
    Slack3_D = SL_D(1,12);
    Excess1_D = SL_D(1,13);
    Excess2_D = SL_D(1,14);

for e=1:14
    SL_E(e)=XE(e);
end

    Slack1_E = SL_E(1,10);
    Slack2_E = SL_E(1,11);
    Slack3_E = SL_E(1,12);
    Excess1_E = SL_E(1,13);
    Excess2_E = SL_E(1,14);

for f=1:14
    SL_F(f)=XF(f);
end

    Slack1_F = SL_F(1,10);
    Slack2_F = SL_F(1,11);
    Slack3_F = SL_F(1,12);
    Excess1_F = SL_F(1,13);
    Excess2_F = SL_F(1,14);

for g=1:14
    SL_G(g)=XG(g);
end

    Slack1_G = SL_G(1,10);
    Slack2_G = SL_G(1,11);
    Slack3_G = SL_G(1,12);
    Excess1_G = SL_G(1,13);
    Excess2_G = SL_G(1,14);

for h=1:14
    SL_H(h)=XH(h);
end

    Slack1_H = SL_H(1,10);
    Slack2_H = SL_H(1,11);
    Slack3_H = SL_H(1,12);
    Excess1_H = SL_H(1,13);
    Excess2_H = SL_H(1,14);

for i=1:14
    SL_I(i)=XI(i);
end

```

```

Slack1_I = SL_I(1,10);
Slack2_I = SL_I(1,11);
Slack3_I = SL_I(1,12);
Excess1_I = SL_I(1,13);
Excess2_I = SL_I(1,14);

```

```
clc;
```

```
%Reports
```

```

disp('===== DEA
MATLAB Solver
=====');
disp('===== Eficiência =          RADIAL          = FOLGAS (SLACKS) ==
FALTAS    ==    ALVOS (TARGETS)    ==');
disp('== DMU == Theta = I1 == I2 == O1 == O2 == O3 = I1 == I2 == O1 == O2 ==
O3 == I1 == I2 == O1 == O2 == O3 ==');
disp('=====
=====');
%Results summary from A
fprintf('== M.Net == %5.4f', FVAL_A);
fprintf(' ==');
fprintf(' %5.2f',Radial1_A);
fprintf(' ==');
fprintf(' %5.2f',Radial2_A);
fprintf(' ==');
fprintf(' %5.2f',Radial3_A);
fprintf(' ==');
fprintf(' %5.2f',Radial4_A);
fprintf(' ==');
fprintf(' %5.2f',Radial5_A);
fprintf(' ==');
fprintf(' %2.0f',inputSlack1_A);
fprintf(' ==');
fprintf(' %4.2f',inputSlack2_A);
fprintf(' ==');
fprintf(' %2.0f ',Target3_A);% this variable was change its semantic
fprintf('==');
fprintf(' %4.0f ',Target4_A);
fprintf(' ==');
fprintf(' %4.0f',Target5_A);
fprintf(' ==');
fprintf(' %4.0f ',Target1_A);
fprintf('==');
fprintf(' %5.1f',Target2_A);
fprintf(' ==');
fprintf(' %3.0f,outputSlack1_A);
fprintf(' ==');
fprintf(' %3.0f,outputSlack2_A);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_A);
fprintf(' ==');
fprintf('\n');
%Results summary from B
fprintf('== Baydenet == %5.4f', FVAL_B);
fprintf(' ==');
fprintf(' %5.2f',Radial1_B);
fprintf('==');
fprintf(' %5.2f',Radial2_B);

```



```

fprintf(' ==');
fprintf('%5.2f',Radial3_B);
fprintf(' ==');
fprintf('%5.0f',Radial4_B);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_B);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_B);
fprintf(' ==');
fprintf('%3.0f,inputSlack2_B);
fprintf(' == ');
fprintf('%2.0f,Target3_B);% this variable was change its semantic
fprintf(' ==');
fprintf(' %2.0f',Target4_B);
fprintf(' ==');
fprintf(' %2.0f',Target5_B);
fprintf(' ==');
fprintf(' %4.0f ',Target1_B);
fprintf(' ==');
fprintf(' %3.0f',Target2_B);
fprintf(' ==');
fprintf('%3.0f,outputSlack1_B);
fprintf(' ==');
fprintf(' %2.0f ',outputSlack2_B);
fprintf(' ==');
fprintf(' %4.0f,outputSlack3_B);
fprintf(' ==');
fprintf('\n');
%Results summary from C
fprintf('== ADLLink == %5.4f, FVAL_C);
fprintf(' ==');
fprintf(' %5.2f',Radial1_C);
fprintf(' ==');
fprintf(' %5.2f',Radial2_C);
fprintf(' ==');
fprintf(' %5.2f',Radial3_C);
fprintf(' ==');
fprintf(' %5.0f',Radial4_C);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_C);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_C);
fprintf(' ==');
fprintf('%3.0f,inputSlack2_C);
fprintf(' ==');
fprintf('%3.0f,Target3_C);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f',Target4_C);
fprintf(' ==');
fprintf('%3.0f,Target5_C);
fprintf(' ==');
fprintf(' %3.0f ',Target1_C);
fprintf(' ==');
fprintf(' %3.0f ',Target2_C);
fprintf(' ==');
fprintf('%3.0f ',outputSlack1_C);
fprintf(' ==');
fprintf(' %2.0f ',outputSlack2_C);
fprintf(' ==');
fprintf(' %5.0f ',outputSlack3_C);

```

```

fprintf(' ==');
fprintf('\n');
%Results summary from D
fprintf('== NetLine == %5.4f,    FVAL_D);
fprintf(' ==');
fprintf('%3.2f,Radial1_D);
fprintf(' ==');
fprintf(' %5.2f,Radial2_D);
fprintf(' ==');
fprintf(' %4.2f,Radial3_D);
fprintf(' ==');
fprintf(' %3.0f,Radial4_D);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_D);
fprintf('==');
fprintf('%2.0f,inputSlack1_D);
fprintf(' ==');
fprintf(' %2.0f ',inputSlack2_D);
fprintf(' ==');
fprintf('%3.0f,Target3_D);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f,Target4_D);
fprintf(' ==');
fprintf(' %2.0f,Target5_D);
fprintf(' ==');
fprintf('%3.0f ',Target1_D);
fprintf('==');
fprintf(' %3.0f ',Target2_D);
fprintf('==');
fprintf(' %2.0f ',outputSlack1_D);
fprintf('==');
fprintf('%2.0f,outputSlack2_D);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_D);
fprintf(' ==');
fprintf('\n');

```

```

%Results summary from E
fprintf('== CEPlug == %5.4f,    FVAL_E);
fprintf(' ==');
fprintf(' %5.2f,Radial1_E);
fprintf(' ==');
fprintf(' %5.2f,Radial2_E);
fprintf(' ==');
fprintf(' %4.2f,Radial3_E);
fprintf(' == ');
fprintf('%4.0f,Radial4_E);
fprintf(' ==');
fprintf(' %5.0f ',Radial5_E);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_E);
fprintf(' ==');
fprintf(' %2.0f,inputSlack2_E);
fprintf(' ==');
fprintf('%3.0f,Target3_E);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %2.0f,Target4_E);
fprintf(' ==');
fprintf(' %2.0f ',Target5_E);
fprintf(' ==');

```

```

fprintf(' %2.0f ',Target1_E);
fprintf('==');
fprintf(' %3.0f ',Target2_E);
fprintf('==');
fprintf('%3.0f,outputSlack1_E);
fprintf(' ==');
fprintf('%3.0f,outputSlack2_E);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_E);
fprintf(' ==');
fprintf('\n');%Results summary from F
fprintf('== HotLink == %5.4f', FVAL_F);
fprintf(' ==');
fprintf(' %5.2f',Radial1_F);
fprintf(' ==');
fprintf(' %5.2f',Radial2_F);
fprintf(' ==');
fprintf('%5.2f,Radial3_F);
fprintf(' ==');
fprintf('%3.2f,Radial4_F);
fprintf(' ==');
fprintf('%5.2f,Radial5_F);
fprintf('==');
fprintf('%2.0f,inputSlack1_F);
fprintf(' ==');
fprintf(' %5.2f,inputSlack2_F);
fprintf(' ==');
fprintf('%3.0f,Target3_F);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %3.0f,Target4_F);
fprintf(' ==');
fprintf('%5.2f,Target5_F);
fprintf('==');
fprintf('%4.2f,Target1_F);
fprintf('==');
fprintf('%5.2f ',Target2_F);
fprintf('==');
fprintf('%3.0f,outputSlack1_F);
fprintf(' ==');
fprintf('%3.0f,outputSlack2_F);
fprintf(' ==');
fprintf(' %5.2f,outputSlack3_F);
fprintf(' ==');
fprintf('\n');

```

```

%Results summary from G
fprintf('== Cz.Net == %5.4f', FVAL_G);
fprintf(' ==');
fprintf(' %5.2f,Radial1_G);
fprintf(' ==');
fprintf(' %3.2f ',Radial2_G);
fprintf('==');
fprintf(' %3.2f,Radial3_G);
fprintf(' ==');
fprintf('%3.2f ',Radial4_G);
fprintf('==');
fprintf('%5.2f,Radial5_G);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_G);
fprintf(' ==');

```

```

fprintf(' %5.2f',inputSlack2_G);
fprintf(' ==');
fprintf('%3.2f',Target3_G);% this variable was change its semantic wick outputSlack
fprintf('==');
fprintf(' %2.0f ',Target4_G);
fprintf('==');
fprintf(' %4.0f',Target5_G);
fprintf(' ==');
fprintf(' %3.0f',Target1_G);
fprintf(' ==');
fprintf(' %5.2f',Target2_G);
fprintf(' ==');
fprintf('%3.2f,outputSlack1_G);
fprintf('==');
fprintf('%3.0f,outputSlack2_G);
fprintf(' ==');
fprintf(' %5.0f',outputSlack3_G);
fprintf(' ==');
fprintf('\n');
%Results summary from H
fprintf('== WiNet == %5.4f, FVAL_H);
fprintf(' ==');
fprintf(' %5.2f',Radial1_H);
fprintf(' ==');
fprintf(' %5.2f',Radial2_H);
fprintf(' ==');
fprintf(' %4.2f',Radial3_H);
fprintf(' ==');
fprintf(' %2.0f',Radial4_H);
fprintf(' ==');
fprintf(' %5.0f',Radial5_H);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_H);
fprintf(' ==');
fprintf(' %2.0f',inputSlack2_H);
fprintf(' ==');
fprintf(' %2.0f ',Target3_H);% this variable was change its semantic wick outputSlack
fprintf('==');
fprintf(' %2.0f',Target4_H);
fprintf(' ==');
fprintf(' %2.0f',Target5_H);
fprintf(' ==');
fprintf(' %3.0f',Target1_H);
fprintf(' ==');
fprintf(' %3.0f ',Target2_H);
fprintf('==');
fprintf('%3.0f,outputSlack1_H);
fprintf(' ==');
fprintf(' %2.0f',outputSlack2_H);
fprintf(' ==');
fprintf(' %5.0f',outputSlack3_H);
fprintf(' ==');
fprintf('\n');%Results summary from I
fprintf('= Throughput = %5.4f, FVAL_I);
fprintf(' ==');
fprintf(' %5.2f',Radial1_I);
fprintf(' ==');
fprintf(' %5.2f',Radial2_I);
fprintf(' ==');
fprintf('%5.2f',Radial3_I);

```

```

fprintf(' ==');
fprintf('%5.2f,Radial4_I);
fprintf(' ==');
fprintf('%5.2f,Radial5_I);
fprintf(' ==');
fprintf('%2.0f,inputSlack1_I);
fprintf(' ==');
fprintf(' %5.2f,inputSlack2_I);
fprintf(' ==');
fprintf(' %2.0f,Target3_I);% this variable was change its semantic wich outputSlack
fprintf(' ==');
fprintf(' %2.0f,Target4_I);
fprintf(' ==');
fprintf('%5.2f,Target5_I);
fprintf(' ==');
fprintf('%5.2f,Target1_I);
fprintf(' ==');
fprintf('%5.2f ',Target2_I);
fprintf(' ==');
fprintf(' %2.0f,outputSlack1_I);
fprintf(' ==');
fprintf(' %2.0f,outputSlack2_I);
fprintf(' ==');
fprintf(' %5.0f,outputSlack3_I);
fprintf(' ==');
fprintf('\n');
disp('=====
=====
=====');
fprintf('\n');

```

% MBF Reports

```

disp('=====
=====');
disp('===== DEA MATLAB Solver - MBF
=====');
disp('===== SOMA (FOLGAS + FALTAS) == FOLGAS (INPUT) == FALTAS
(OUTPUT) ==');
disp('== DMU == I1 == I2 == O1 == O2 == O3 ==');
disp('=====
=====');
fprintf('== M.Net ==');
fprintf(' %5.2f,-SLACKA);
fprintf(' ==');
fprintf(' %5.2f,Excess1_A);
fprintf(' ==');
fprintf(' %5.2f, Excess2_A);
fprintf(' ==');
fprintf(' %5.2f,Slack1_A);
fprintf(' ==');
fprintf(' %5.2f,Slack2_A);
fprintf(' ==');
fprintf(' %5.2f,Slack3_A);
fprintf(' ==');
fprintf('\n');
fprintf('== Baydenet ==');
fprintf(' %5.2f ',-SLACKB);
fprintf(' ==');
fprintf(' %5.2f,Excess1_B);
fprintf(' ==');

```

```

fprintf(' %5.2f, Excess2_B);
fprintf(' ==');
fprintf(' %5.2f,Slack1_B);
fprintf(' ==');
fprintf(' %5.2f,Slack2_B);
fprintf(' ==');
fprintf(' %5.2f,Slack3_B);
fprintf(' ==');
fprintf('\n');
fprintf('== ADLLink ==');
fprintf(' %5.2f ',-SLACKC);
fprintf(' ==');
fprintf(' %5.2f,Excess1_C);
fprintf(' ==');
fprintf(' %5.2f, Excess2_C);
fprintf(' ==');
fprintf(' %5.2f,Slack1_C);
fprintf(' ==');
fprintf(' %5.2f,Slack2_C);
fprintf(' ==');
fprintf(' %5.2f,Slack3_C);
fprintf(' ==');
fprintf('\n');
fprintf('== NetLine ==');
fprintf(' %5.2f ',-SLACKD);
fprintf(' ==');
fprintf(' %5.2f,Excess1_D);
fprintf(' ==');
fprintf(' %5.2f, Excess2_D);
fprintf(' ==');
fprintf(' %5.2f,Slack1_D);
fprintf(' ==');
fprintf(' %5.2f,Slack2_D);
fprintf(' ==');
fprintf(' %5.2f,Slack3_D);
fprintf(' ==');
fprintf('\n');
fprintf('== CEPlug ==');
fprintf(' %5.2f ',-SLACKE);
fprintf(' ==');
fprintf(' %5.2f,Excess1_E);
fprintf(' ==');
fprintf(' %5.2f, Excess2_E);
fprintf(' ==');
fprintf(' %5.2f,Slack1_E);
fprintf(' ==');
fprintf(' %5.2f,Slack2_E);
fprintf(' ==');
fprintf(' %5.2f,Slack3_E);
fprintf(' ==');
fprintf('\n');
fprintf('== HotLink ==');
fprintf(' %5.2f ',-SLACKF);
fprintf(' ==');
fprintf(' %5.2f,Excess1_F);
fprintf(' ==');
fprintf(' %5.2f, Excess2_F);
fprintf(' ==');
fprintf(' %5.2f,Slack1_F);
fprintf(' ==');

```

```

fprintf(' %5.2f,Slack2_F);
fprintf(' ==');
fprintf(' %5.2f,Slack3_F);
fprintf(' ==');
fprintf('\n');
fprintf('== Cz.Net ==');
fprintf(' %5.2f,-SLACKG);
fprintf(' ==');
fprintf(' %5.2f,Excess1_G);
fprintf(' ==');
fprintf(' %5.2f, Excess2_G);
fprintf(' ==');
fprintf(' %5.2f,Slack1_G);
fprintf(' ==');
fprintf(' %5.2f,Slack2_G);
fprintf(' ==');
fprintf(' %5.2f,Slack3_G);
fprintf(' ==');
fprintf('\n');
fprintf('== WiNet ==');
fprintf(' %5.2f ',-SLACKH);
fprintf(' ==');
fprintf(' %5.2f,Excess1_H);
fprintf(' ==');
fprintf(' %5.2f, Excess2_H);
fprintf(' ==');
fprintf(' %5.2f,Slack1_H);
fprintf(' ==');
fprintf(' %5.2f,Slack2_H);
fprintf(' ==');
fprintf(' %5.2f,Slack3_H);
fprintf(' ==');
fprintf('\n');
fprintf('== Throughput ==');
fprintf(' %5.2f,-SLACKI);
fprintf(' ==');
fprintf(' %5.2f,Excess1_I);
fprintf(' ==');
fprintf(' %5.2f, Excess2_I);
fprintf(' ==');
fprintf(' %5.2f,Slack1_I);
fprintf(' ==');
fprintf(' %5.2f,Slack2_I);
fprintf(' ==');
fprintf(' %5.2f,Slack3_I);
fprintf(' ==');
fprintf('\n');
disp('=====
=====');

```

% Peers and Benchmarks Reports

```

disp('=====
=====');
disp('== DMU == Parceiro(s) de Excelência == Lambdas ==');
disp('=====
=====');
fprintf('== 1 - M.Net ==');
fprintf(' %5.2f,LambdaA_index);
fprintf(' ==');

```

```

fprintf(' %5.2f',lambdaA);
fprintf('\n');
fprintf('== 2 - Baydenet ==');
fprintf(' %5.2f',LambdaB_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaB);
fprintf('\n');
fprintf('== 3 - ADLLink ==');
fprintf(' %5.2f',LambdaC_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaC);
fprintf('\n');
fprintf('== 4 - NetLine ==');
fprintf(' %5.2f',LambdaD_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaD);
fprintf('\n');
fprintf('== 5 - CEPlug ==');
fprintf(' %5.2f',LambdaE_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaE);
fprintf('\n');
fprintf('== 6 - HotLink ==');
fprintf(' %5.2f',LambdaF_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaF);
fprintf('\n');
fprintf('== 7 - CzNet ==');
fprintf(' %5.2f',LambdaG_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaG);
fprintf('\n');
fprintf('== 8 - WiNet ==');
fprintf(' %5.2f',LambdaH_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaH);
fprintf('\n');
fprintf('== 9 - Throughput ==');
fprintf(' %5.2f',LambdaI_index);
fprintf('      ==');
fprintf(' %5.2f',lambdaI);
fprintf('\n');
disp('=====');
=====);

```

% Graphics

```

x_axes = [FVAL_A,FVAL_B,FVAL_C,FVAL_D,FVAL_E,FVAL_F,FVAL_G,FVAL_H,FVAL_I];
figure1=figure;
barh(x_axes(1,1:9), 'DisplayName', {'A','B','C','D','E','F','G','H','T'});
title('Eficiência das DMUs');
set(gca,'YTickLabel', {'A','B','C','D','E','F','G','H','T'})

```


% DEA - BCC - RVE - Multiplier - Input Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;  
clear;  
b = zeros(9,1); % constraints
```

```
DMU= [2 2 1400 -163 -215 +1;  
      5 50 50000 -2000 -233 +1;  
      5 30 10000 -375 -220 +1;  
      2 125 40000 -1210 -245 +1;  
      4 20 15000 -230 -223 +1;  
      2 30 20000 -220 -235 +1;  
      1 18 4000 -190 -240 +1;  
      3 50 35000 -300 -230 +1;  
      2 30 20000 -205 -235 +1;];
```

```
f_A = [2 2 1400 0 0 1]; % objective function  
f_B = [5 50 50000 0 0 1];  
f_C = [5 30 10000 0 0 1];  
f_D = [2 125 40000 0 0 1];  
f_E = [4 20 15000 0 0 1];  
f_F = [2 30 20000 0 0 1];  
f_G = [1 18 4000 0 0 1];  
f_H = [3 50 35000 0 0 1];  
f_I = [2 30 20000 0 0 1];  
DMUA_Aeq = [0 0 0 163 215 0];  
DMUB_Aeq = [0 0 0 2000 233 0];  
DMUC_Aeq = [0 0 0 375 220 0];  
DMUD_Aeq = [0 0 0 1210 245 0];  
DMUE_Aeq = [0 0 0 230 223 0];  
DMUF_Aeq = [0 0 0 220 235 0];  
DMUG_Aeq = [0 0 0 190 240 0];  
DMUH_Aeq = [0 0 0 300 230 0];  
DMUI_Aeq = [0 0 0 205 235 0];
```

```
Beq = 1;
```

```
lb = zeros(6,1);  
clc;
```

% Calculation of Phi

```
[XA,FVAL_A,EXITFLAG,OUTPUT]=linprog(-f_A,DMU,b,DMUA_Aeq,Beq,lb);  
[XB,FVAL_B,EXITFLAG,OUTPUT]=linprog(-f_B,DMU,b,DMUB_Aeq,Beq,lb);  
[XC,FVAL_C,EXITFLAG,OUTPUT]=linprog(-f_C,DMU,b,DMUC_Aeq,Beq,lb);  
[XD,FVAL_D,EXITFLAG,OUTPUT]=linprog(-f_D,DMU,b,DMUD_Aeq,Beq,lb);  
[XE,FVAL_E,EXITFLAG,OUTPUT]=linprog(-f_E,DMU,b,DMUE_Aeq,Beq,lb);  
[XF,FVAL_F,EXITFLAG,OUTPUT]=linprog(-f_F,DMU,b,DMUF_Aeq,Beq,lb);  
[XG,FVAL_G,EXITFLAG,OUTPUT]=linprog(-f_G,DMU,b,DMUG_Aeq,Beq,lb);  
[XH,FVAL_H,EXITFLAG,OUTPUT]=linprog(-f_H,DMU,b,DMUH_Aeq,Beq,lb);
```

```

[XI,FVAL_I,EXITFLAG,OUTPUT]=linprog(-f_I,DMU,b,DMUI_Aeq,Beq,lb);

% BCC/ M / I

f = [0 0 0 0 0 0 0 0 0 1];
b_A = [-163 -215 0 0 0];

DMU_A = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -2; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -1400;]; % Output_3

b_B = [-2000 -233 0 0 0];

DMU_B = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -50000;]; % Output_3

b_C = [-375 -220 0 0 0];

DMU_C = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -10000;]; % Output_3

b_D = [-1210 -245 0 0 0];

DMU_D = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -125; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -40000;]; % Output_3

b_E = [-230 -223 0 0 0];

DMU_E = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -4; % Output_1
2 50 30 125 20 30 18 50 30 -20; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -15000;]; % Output_3

b_F = [-220 -235 0 0 0];

DMU_F = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

b_G = [-190 -240 0 0 0];

DMU_G = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -1; % Output_1
2 50 30 125 20 30 18 50 30 -18; % Output_2

```

```

1400 50000 10000 40000 15000 20000 4000 35000 20000 -4000;]; % Output_3

b_H = [-300 -230 0 0 0];

DMU_H = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -3; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -35000;]; % Output_3

b_I = [-205 -235 0 0 0];

DMU_I = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

lb= zeros(9,1);
% Technical Efficiency CCR
[lambda_A,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f,-DMU_A,-b_A,[],[],lb);
CCR_A=-1/FVAL_1;

[lambda_B,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f,-DMU_B,-b_B,[],[],lb);
CCR_B=-1/FVAL_2;

[lambda_C,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f,-DMU_C,-b_C,[],[],lb);
CCR_C=-1/FVAL_3;

[lambda_D,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f,-DMU_D,-b_D,[],[],lb);
CCR_D=-1/FVAL_4;

[lambda_E,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f,-DMU_E,-b_E,[],[],lb);
CCR_E=-1/FVAL_5;

[lambda_F,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f,-DMU_F,-b_F,[],[],lb);
CCR_F=-1/FVAL_6;

[lambda_G,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f,-DMU_G,-b_G,[],[],lb);
CCR_G=-1/FVAL_7;

[lambda_H,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f,-DMU_H,-b_H,[],[],lb);
CCR_H=-1/FVAL_8;

[lambda_I,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f,-DMU_I,-b_I,[],[],lb);
CCR_I=-1/FVAL_9;

```

% Technical Efficiency versus Scale Efficiency - Reports

```

clc;

disp('=====');
);
disp('          Efficiencies');

disp('=====');
);
disp('DMU      -      CCR      -      BCC      ');

```

```

disp('=====')
);
fprintf('== A ==      %5.4f,CCR_A);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_A);
fprintf('\n');
fprintf('== B ==      %5.4f,CCR_B);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_B);
fprintf('\n');
fprintf('== C ==      %5.4f,CCR_C);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_C);
fprintf('\n');
fprintf('== D ==      %5.4f,CCR_D);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_D);
fprintf('\n');
fprintf('== E ==      %5.4f,CCR_E);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_E);
fprintf('\n');
fprintf('== F ==      %5.4f,CCR_F);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_F);
fprintf('\n');
fprintf('== G ==      %5.4f,CCR_G);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_G);
fprintf('\n');
fprintf('== H ==      %5.4f,CCR_H);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_H);
fprintf('\n');
fprintf('== I ==      %5.4f,CCR_I);
fprintf(' ==');
fprintf('      %5.4f, -FVAL_I);
fprintf('\n');

disp('=====')
);

```

% BCC – Graphic

```

x_axes = [-FVAL_A,-FVAL_B,-FVAL_C,-FVAL_D,-FVAL_E,-FVAL_F,-FVAL_G,-FVAL_H,-
FVAL_I];
figure1=figure;
barh(x_axes(1,1:9), 'DisplayName',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'});
title('Eficiência das DMUs - BCC');
set(gca,'YTickLabel',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'})

```

% CCR – Graphic

```

X_axes = [CCR_A,CCR_B,CCR_C,CCR_D,CCR_E,CCR_F,CCR_G,CCR_H,CCR_I];
figure2=figure;
barh(X_axes(1,1:9), 'DisplayName',
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'});

```

```
title('Eficiência das DMUs - CCR');  
set(gca,'YTickLabel',  
{'M.Net','Baydenet','ADLLink','NetLine','CEPlug','HotLink','Cz.Net','WiNet','Throughput'})
```

% CCR - Envelopment – Output – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

clear;

clc;

f = [0 0 0 0 0 0 0 0 1];

b_A = [-163 -215 0 0 0];

DMU_A = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -2; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -1400;]; % Output_3

b_B = [-2000 -233 0 0 0];

DMU_B = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -50000;]; % Output_3

b_C = [-375 -220 0 0 0];

DMU_C = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0 ; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -5; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -10000;]; % Output_3

b_D = [-1210 -245 0 0 0];

DMU_D = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -125; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -40000;]; % Output_3

b_E = [-230 -223 0 0 0];

DMU_E = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -4; % Output_1
2 50 30 125 20 30 18 50 30 -20; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -15000;]; % Output_3

b_F = [-220 -235 0 0 0];

DMU_F = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

b_G = [-190 -240 0 0 0];

DMU_G = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -1; % Output_1

```

2 50 30 125 20 30 18 50 30 -18; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -4000;]; % Output_3

b_H = [-300 -230 0 0 0];

DMU_H = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -3; % Output_1
2 50 30 125 20 30 18 50 30 -50; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -35000;]; % Output_3

b_I = [-205 -235 0 0 0];

DMU_I = [-163 -2000 -375 -1210 -230 -220 -190 -300 -205 0; % Input_1
-215 -233 -220 -245 -223 -235 -240 -230 -235 0; % Input_2
2 5 5 2 4 2 1 3 2 -2; % Output_1
2 50 30 125 20 30 18 50 30 -30; % Output_2
1400 50000 10000 40000 15000 20000 4000 35000 20000 -20000;]; % Output_3

lb= zeros(9,1);

% Calculation of Technical Efficiency
[XA,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f,-DMU_A,-b_A,[],[],lb);
disp('A eficácia da DMU_A é:');
disp(-1/FVAL_1);

[XB,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f,-DMU_B,-b_B,[],[],lb);
disp('A eficácia da DMU_B é:');
disp(-1/FVAL_2);

[XC,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f,-DMU_C,-b_C,[],[],lb);
disp('A eficácia da DMU_C é:');
disp(-1/FVAL_3);

[XD,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f,-DMU_D,-b_D,[],[],lb);
disp('A eficácia da DMU_D é:');
disp(-1/FVAL_4);

[XE,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f,-DMU_E,-b_E,[],[],lb);
disp('A eficácia da DMU_E é:');
disp(-1/FVAL_5);

[XF,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f,-DMU_F,-b_F,[],[],lb);
disp('A eficácia da DMU_F é:');
disp(-1/FVAL_6);

[XG,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f,-DMU_G,-b_G,[],[],lb);
disp('A eficácia da DMU_G é:');
disp(-1/FVAL_7);

[XH,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f,-DMU_H,-b_H,[],[],lb);
disp('A eficácia da DMU_H é:');
disp(-1/FVAL_8);

[XI,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f,-DMU_I,-b_I,[],[],lb);
disp('A eficácia da DMU_I é:');
disp(-1/FVAL_9);

```

% DEA - CCR - Multipliers Mode - Input Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;  
clear;  
b = [ 0; 0; 0; 0; 0; 0; 0; 0; 0; 0;]; % constraints
```

```
Beq = 1;
```

```
DMU= [2 2 1400 -163 -215;  
      5 50 50000 -2000 -233;  
      5 30 10000 -375 -220;  
      2 125 40000 -1210 -245;  
      4 20 15000 -230 -223;  
      2 30 20000 -220 -235;  
      1 18 4000 -190 -240;  
      3 50 35000 -300 -230;  
      2 30 20000 -205 -235;];
```

```
f_A = [2 2 1400 0 0]; % objective function
```

```
DMUA_Aeq = [0 0 0 163 215;];
```

```
f_B = [5 50 50000 0 0];
```

```
DMUB_Aeq = [0 0 0 2000 233];
```

```
f_C = [ 5; 30; 10000; 0; 0;];
```

```
DMUC_Aeq = [ 0 0 0 375 220;];
```

```
f_D = [2;125; 40000; 0;0;];
```

```
DMUD_Aeq=[ 0 0 0 1210 245;];
```

```
f_E = [4; 20; 15000; 0; 0;];
```

```
DMUE_Aeq=[ 0 0 0 230 223;];
```

```
f_F = [2; 30; 20000; 0; 0;];
```

```
DMUF_Aeq = [ 0 0 0 220 235;];
```

```
f_G = [1 18 4000 0 0;];
```

```
DMUG_Aeq = [ 0 0 0 190 240];
```

```
f_H = [ 3; 50; 35000; 0; 0;];
```

```
DMUH_Aeq = [ 0 0 0 300 230;];
```

```
f_I = [ 2; 30; 20000; 0; 0;];
```

```
DMUI_Aeq = [ 0 0 0 205 235;];
```

```
lb =zeros(5,1);
```

```
% Calculation of Technical Efficiency
```

```
[XA,FVAL_1,EXITFLAG,OUTPUT]=linprog(-f_A,DMU,b,DMUA_Aeq,Beq,lb);
```



```
disp('A eficácia da DMU_A é:');
disp(-FVAL_1);
```

```
[XB,FVAL_2,EXITFLAG,OUTPUT]=linprog(-f_B,DMU,b,DMUB_Aeq,Beq,lb);
disp('A eficácia da DMU_B é:');
disp(-FVAL_2);
```

```
[XC,FVAL_3,EXITFLAG,OUTPUT]=linprog(-f_C,DMU,b,DMUC_Aeq,Beq,lb);
disp('A eficácia da DMU_C é:');
disp(-FVAL_3);
```

```
[XD,FVAL_4,EXITFLAG,OUTPUT]=linprog(-f_D,DMU,b,DMUD_Aeq,Beq,lb);
disp('A eficácia da DMU_D é:');
disp(-FVAL_4);
```

```
[XE,FVAL_5,EXITFLAG,OUTPUT]=linprog(-f_E,DMU,b,DMUE_Aeq,Beq,lb);
disp('A eficácia da DMU_E é:');
disp(-FVAL_5);
```

```
[XF,FVAL_6,EXITFLAG,OUTPUT]=linprog(-f_F,DMU,b,DMUF_Aeq,Beq,lb);
disp('A eficácia da DMU_F é:');
disp(-FVAL_6);
```

```
[XG,FVAL_7,EXITFLAG,OUTPUT]=linprog(-f_G,DMU,b,DMUG_Aeq,Beq,lb);
disp('A eficácia da DMU_G é:');
disp(-FVAL_7);
```

```
[XH,FVAL_8,EXITFLAG,OUTPUT]=linprog(-f_H,DMU,b,DMUH_Aeq,Beq,lb);
disp('A eficácia da DMU_H é:');
disp(-FVAL_8);
```

```
[XI,FVAL_9,EXITFLAG,OUTPUT]=linprog(-f_I,DMU,b,DMUI_Aeq,Beq,lb);
disp('A eficácia da DMU_I é:');
disp(-FVAL_9);
```

```
theta=[-FVAL_1 -FVAL_2 -FVAL_3 -FVAL_4 -FVAL_5 -FVAL_6 -FVAL_7 -FVAL_8 -FVAL_9];
theta_A=-FVAL_1;
theta_B=-FVAL_2;
theta_C=-FVAL_3;
theta_D=-FVAL_4;
theta_E=-FVAL_5;
theta_F=-FVAL_6;
theta_G=-FVAL_7;
theta_H=-FVAL_8;
theta_I=-FVAL_9;
```

% Slacks

```
if (theta_A||1)
    Input1=DMU(1,4);
    Input2=DMU(1,5);
    disp('Valores imutáveis da entrada da DMU_A:');
    disp('Input1:' +Input1);
    disp('Input2:' +Input2);
end
if (theta_A||1)
    Input1=DMU(1,4)*theta_A;
    Input2=DMU(1,5)*theta_A;
    disp('Novos valores de entrada da DMU_A:');
```

```

        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_B||1)
        Input1=DMU(2,4);
        Input2=DMU(2,5);
        disp('Valores imutáveis da entrada da DMU_B:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_B||1)
        Input1=DMU(2,4)*theta_B;
        Input2=DMU(2,5)*theta_B;
        disp('Novos valores de entrada da DMU_B:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_C||1)
        Input1=DMU(3,4);
        Input2=DMU(3,5);
        disp('Valores imutáveis da entrada da DMU_C:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_C||1)
        Input1=DMU(2,4)*theta_C;
        Input2=DMU(2,5)*theta_C;
        disp('Novos valores de entrada da DMU_C:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_D||1)
        Input1=DMU(4,4);
        Input2=DMU(4,5);
        disp('Valores imutáveis da entrada da DMU_D:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
    if (theta_D||1)
        Input1=DMU(4,4)*theta_D;
        Input2=DMU(4,5)*theta_D;
        disp('Novos valores de entrada da DMU_D:');
        disp('Input1:');
        disp(Input1);
        disp('Input2:');
        disp(Input2);
    end
end

```

```

if (theta_E||1)
    Input1=DMU(5,4);
    Input2=DMU(5,5);
    disp('Valores imutáveis da entrada da DMU_E:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_E||1)
    Input1=DMU(5,4)*theta_E;
    Input2=DMU(5,5)*theta_E;
    disp('Novos valores de entrada da DMU_E:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_F||1)
    Input1=DMU(6,4);
    Input2=DMU(6,5);
    disp('Valores imutáveis da entrada da DMU_F:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_F||1)
    Input1=DMU(6,4)*theta_F;
    Input2=DMU(6,5)*theta_F;
    disp('Novos valores de entrada da DMU_F:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_G||1)
    Input1=DMU(7,4);
    Input2=DMU(7,5);
    disp('Valores imutáveis da entrada da DMU_G:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_G||1)
    Input1=DMU(7,4)*theta_G;
    Input2=DMU(7,5)*theta_G;
    disp('Novos valores de entrada da DMU_G:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end

if (theta_H||1)
    Input1=DMU(8,4);
    Input2=DMU(8,5);
    disp('Valores imutáveis da entrada da DMU_H:');
    disp('Input1:');

```

```

    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_H||1)
    Input1=DMU(8,4)*theta_H;
    Input2=DMU(8,5)*theta_H;
    disp('Novos valores de entrada da DMU_H:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end

if (theta_I||1)
    Input1=DMU(9,4);
    Input2=DMU(9,5);
    disp('Valores imutáveis da entrada da DMU_I:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
if (theta_I||1)
    Input1=DMU(9,4)*theta_I;
    Input2=DMU(9,5)*theta_I;
    disp('Novos valores de entrada da DMU_I:');
    disp('Input1:');
    disp(Input1);
    disp('Input2:');
    disp(Input2);
end
end

```

% DEA - CCR - Multiplier - Output Oriented – MATLAB Code
% Developed by Francisco Daladier Marques Junior
% contact: daladierjr@ifpb.edu.br

```
clc;
b = [ 1; 0; 0; 0; 0; 0; 0; 0; 0;]; % constraints

%DMU_A = [ 2 0 0; 2 -6 -14; 4 -8 -16; 3 -12 -12; 2 -6 -4; 1 -6 -3; 3 -15 -3];
DMU= [2 2 1400 -163 -215;
      5 50 50000 -2000 -233;
      5 30 10000 -375 -220;
      2 125 40000 -1210 -245;
      4 20 15000 -230 -223;
      2 30 20000 -220 -235;
      1 18 4000 -190 -240;
      3 50 35000 -300 -230;
      2 30 20000 -205 -235;];
f_A = [0; 0; 0; 163; 215;]; % objective function

DMUA_Aeq = [2 2 1400 0 0]
DMUA_Beq = [1]

f_B = [ 0; 0; 0; 2000; 233;];
DMUB_Aeq = [5 50 50000 0 0];
DMUB_Beq = [1];

f_C = [ 0; 0; 0; 375; 220;];

DMUC_Aeq = [5 30 10000 0 0]
DMUC_Beq = [1]

f_D = [0; 0; 0; 1210; 245;];
DMUD_Aeq = [2 125 40000 0 0];
DMUD_Beq = [1];

f_E = [0; 0; ; 0; 230; 223;];

DMUE_Aeq = [4 20 15000 0 0];
DMUE_Beq = [1];

f_F = [0; 0; 0; 220; 235;];
DMUF_Aeq = [2 30 20000 0 0];
DMUF_Beq = [1]

f_G = [ 0; 0; 0; 190; 240;];
DMUG_Aeq = [1 18 4000 0 0];
DMUG_Beq = [1];

f_H = [ 0; 0; 0; 300; 230;];
DMUH_Aeq = [3 50 35000 0 0];
DMUH_Beq = [1]

f_I = [ 0; 0; 0; 205; 235;];
DMUI_Aeq = [2 30 20000 0 0];
DMUI_Beq = [1]

lb = zeros(5,1);
```

```
clc;
```

```
% Calculation of Technical
```

```
[X_A,FVAL_A,EXITFLAG,OUTPUT,lambdaA]=linprog(f_A,DMU,b,DMUA_Aeq,DMUB_Beq,lb);  
disp('A eficácia da DMU_A é:');  
disp(1/FVAL_A);
```

```
[X_B,FVAL_B,EXITFLAG,OUTPUT,lambdaB]=linprog(f_B,DMU,b,DMUB_Aeq,DMUB_Beq,lb);  
disp('A eficácia da DMU_B é:');  
disp(1/FVAL_B);
```

```
[X_C,FVAL_C,EXITFLAG,OUTPUT,lambdaC]=linprog(f_C,DMU,b,DMUC_Aeq,DMUC_Beq,lb);  
disp('A eficácia da DMU_C é:');  
disp(1/FVAL_C);
```

```
[X_D,FVAL_D,EXITFLAG,OUTPUT,lambdaD]=linprog(f_D,DMU,b,DMUD_Aeq,DMUD_Beq,lb);  
disp('A eficácia da DMU_D é:');  
disp(1/FVAL_D);
```

```
[X_E,FVAL_E,EXITFLAG,OUTPUT,lambdaE]=linprog(f_E,DMU,b,DMUE_Aeq,DMUE_Beq,lb);  
disp('A eficácia da DMU_E é:');  
disp(1/FVAL_E);
```

```
[X_F,FVAL_F,EXITFLAG,OUTPUT,lambdaF]=linprog(f_F,DMU,b,DMUF_Aeq,DMUF_Beq,lb);  
disp('A eficácia da DMU_F é:');  
disp(1/FVAL_F);
```

```
[X_G,FVAL_G,EXITFLAG,OUTPUT,lambdaG]=linprog(f_G,DMU,b,DMUG_Aeq,DMUG_Beq,lb);  
disp('A eficácia da DMU_G é:');  
disp(1/FVAL_G);
```

```
[X_H,FVAL_H,EXITFLAG,OUTPUT,lambdaH]=linprog(f_H,DMU,b,DMUH_Aeq,DMUH_Beq,lb);  
disp('A eficácia da DMU_H é:');  
disp(1/FVAL_H);
```

```
[X_I,FVAL_I,EXITFLAG,OUTPUT,lambdaI]=linprog(f_I,DMU,b,DMUI_Aeq,DMUI_Beq,lb);  
disp('A eficácia da DMU_I é:');  
disp(1/FVAL_I);
```