TÂNIA CORDEIRO LINDBECK DA SILVA

O PROBLEMA DE PROGRAMAÇÃO DE HORÁRIOS DE TRABALHO CONSIDERANDO PREFERÊNCIAS E HIERARQUIA: APLICAÇÃO A UMA ESCALA DE PLANTÃO DE MILITARES

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências.
Programa de Pós-Graduação em Métodos Numéricos em Engenharia – Programação Matemática,
Setores de Tecnologia e de Ciências Exatas,
Universidade Federal do Paraná.
Orientadora: Profª Drª Maria Teresinha Arns Steiner
Co-orientador: Prof. M.Sc. Arinei Carlos Lindbeck da Silva

CURITIBA

2002

TÂNIA CORDEIRO LINDBECK DA SILVA

O PROBLEMA DE PROGRAMAÇÃO DE HORÁRIOS DE TRABALHO CONSIDERANDO PREFERÊNCIAS E HIERARQUIA: APLICAÇÃO A UMA ESCALA DE PLANTÃO DE MILITARES

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela Comissão formada pelos professores:

Orientadora: Maria Teresinha Arns Steiner

Profª Drª Maria Teresinha Arns Steiner
Departamento de Matemática, UFPR

Prof. Dr. Celso Carnieri
Programa de Pós-Graduação em Métodos Numéricos em Engenharia

Marco A. Mazzarotto

Prof. Dr. Marco André Mazzarotto
Departamento de Matemática, UTP

Curitiba, 25 de julho de 2002

Ao meu marido

AGRADECIMENTOS

Aos meus pais, que estão sempre apoiando e incentivando o crescimento tanto profissional quanto espiritual.

Ao meu marido, que esteve ao meu lado, e muitas vezes à frente para derrubar as barreiras e tentar tornar o caminho um pouco mais fácil para nossa família.

À minha filha Lydia que seguiu seu caminho de maneira correta, às vezes sem poder contar muito comigo, mas com a certeza de que estive ao seu lado.

Ao meu filho André, que veio alegrar mais ainda nossos dias.

Aos professores, que estiveram dispostos em sanar dúvidas que surgiam, e ensinar com muita clareza e objetividade.

Aos colegas de turma, e em especial, aos colegas que se tornaram grandes amigos.

Aos funcionários do CESEC, que colaboraram com muito carinho e atenção sempre que foi necessário.

À Força Aérea Brasileira pelo fornecimento dos dados necessários ao desenvolvimento do trabalho.

À Escola Técnica da Universidade Federal do Paraná pela concessão da licença para a realização do curso e aos colegas pela colaboração e companheirismo.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

RESUMO

O objetivo deste trabalho é resolver o problema de escala de soldados da aeronáutica para o serviço de plantão, chamado serviço de guarda, procurando-se facilitar o trabalho do responsável pela confecção dessas escalas, que o faz manualmente. Para realizar o trabalho, primeiramente foi necessário compreender a hierarquia militar, pesquisando sobre leis e normas militares, e sobre as funções dos militares. Em seguida, como se propõe a modelar o problema como um PPLI, fez-se uma revisão sobre os modelos gerais de Programação Linear Inteira Binária e de alguns métodos específicos para resolução de destes problemas como alguns trabalhos realizados em relação a *rostering* e *scheduling*. Para resolver o PPLIB modelado para a escala de plantão dos militares foi desenvolvido um programa em *Visual Basic* para estabelecer as regras e parâmetros do problema, e utilizado o *software LINGO* para a resolução do mesmo. Para compreensão de como utilizar o *LINGO* dentro de um programa executável foi descrito um exemplo utilizando este *software*. Os resultado apresentados foram analisados em relação ao tempo para obter uma lista de escalas; o quanto foi possível atender às preferências dos militares e, no caso de alterações, como possibilitar as mudanças de escala para o menor número de militares, sempre respeitando a hierarquia. A grande maioria dos militares considerados teve as suas preferências atendidas. Vários testes foram executados considerando diferentes quantidades de militares e diferentes demandas diárias, através de simulações, e foi possível observar a validade da formulação através de aplicações práticas, comparando com tabelas já existentes. Foram efetuadas variações nos pesos atribuídos à antiguidade de cada militar e foi observado que a diferença de pesos é significativa somente quando a quantidade de dias para escolha de cada militar é próxima da demanda necessária deste militar. A idéia de ter-se militares de "sobreaviso" para cobrir as faltas, quando o número de militares disponíveis é muito pequeno torna-se inviável, pois não existe folga suficiente para o atendimento da demanda criada. Neste caso a solução do problema é feita pelo modelo sem sobreaviso e no caso de faltas o modelo é "executado" novamente para o período restante; no entanto os resultados apresentados (movimentação de muitos militares para outros dias diferentes dos originalmente indicados) não foram bons, pelo fato das alterações serem muito grandes. Com um número suficiente de militares, porém, o "sobreaviso" mostrou-se extremamente interessante para o gerenciamento do pessoal, evitando atropelos de chamada de novos militares nos dias de faltas.

ABSTRACT

The objective of this work is to solve the aeronautic soldier scale problem for the duty call, facilitating the work of the responsible for the confection of these scales; which is nowadays made by hand. First it was necessary to understand the military hierarchy, searching on laws and military norms, and the functions of the military. Since it has been chosen to consider the problem an Integer Programming, a revision was made upon the general models of Zero-one Linear Programming and some specific methods for the resolution of these problems as well as some works carried through in rostering and scheduling. To solve the Zero-one Linear Programming shape for the military duty scale a program in *Visual Basic* was developed to establish the rules and parameters for the problem using the LINGO software for the resolution of the problem. For the understanding of how to use the LINGO software inside an executable program an example was described using this software. The result presented had been analyzed in relation to the time to get a list of scales; respecting as long as it was possible the preferences of the military, and in the case of alterations to turn possible the scale changes so that the lesser number of military on duty, always respecting the hierarchy. The great majority of the military preferences had been taken care of. Some tests had been executed considering different amounts of military and different daily demands; through simulation it was possible to observe the validity of the formulation through practical applications, if compared with the existing tables. Variations in the weights attributed to the time of service of each military man had been effected and were observed that the difference of weights is significant only when the amount of days for choice of each military man is next to the necessary demand of this military man. The idea of having cautioned military men to cover the lacks, when the number of available military man is very small the software becomes unable to perform the task; therefore does not are enough recess for the attendance of the demand. In this case the solution of the problem is made by the cautioned military men model, and in the case of lacks the model is remade for the remaining period; however the presented results (movement of many military for other different days from the originally indicated) had not been good, for the fact that there are many alterations. However if there are enough number of military men, on wait the program revealed as being extremely interesting for the management of the staff, improving the call of new military men for the days of lacks.

CAPÍTULO I

1. INTRODUÇÃO

1.1 OBJETIVO DO TRABALHO

O que se propõe neste trabalho é abordar o problema de escala de funcionários, considerando preferências e hierarquia, aplicado à escala de soldados da aeronáutica para o serviço de plantão, chamado serviço de guarda. Procura-se facilitar o trabalho do responsável pela confecção dessas escalas, que é feita no momento de forma manual consumindo muito tempo e, além disso, melhorar o nível de satisfação dos servidores.

O objetivo para o caso em estudo é estabelecer os dias de serviço de guarda de cada soldado, obedecendo às regras da hierarquia militar.

São apresentados neste trabalho também uma revisão sobre algoritmos para solução de modelos de Programação Linear Inteira Binária e alguns métodos específicos para resolução de problemas de escala de funcionários (*manpower scheduling).*

Finalmente, é mostrado como utilizar o *LINGO (Language for Interactive General Optimizer)* dentro de um programa executável para resolver problemas de Programação Linear.

1.2 IMPORTÂNCIA DO TRABALHO

O planejamento da escala de serviço de plantão (ou serviço de guarda) deve ser feito mensalmente em função das férias dos servidores, e como é feito manualmente, o auxílio de técnicas de Pesquisa Operacional vem facilitar e melhorar este serviço de planejamento, obtendo-se uma solução ótima, de forma eficiente e automática.

O serviço de guarda influi diretamente na vida do soldado, pois dias inteiros (manhã e noite) serão dedicados ao serviço militar. Com a possibilidade de escolha dos dias de serviço e a grande chance que esses dias sejam efetivamente os de plantão, o soldado poderá programar melhor a sua vida pessoal.

1.3 Limitação do Trabalho

Com a alteração do número de militares devido às férias e possíveis licenças, e com a possibilidade de escolhas dos dias de plantão por parte dos militares, o planejamento das escalas deve ser mensal, considerando os militares disponíveis para o período, o que pode vir a dificultar um pouco a coleta de dados, limitando o trabalho.

1.4 Estrutura do Trabalho

São descritas a seguir, as funções e a hierarquia militar nas três Armas – Exército, Marinha e Aeronáutica – e o problema de formação de escalas para o Serviço de Plantão na Aeronáutica, especificamente para os Soldados de Primeira e Segunda Classe, no Capítulo II.

No Capítulo III são apresentados alguns modelos e métodos para solução de problemas de Programação Linear Inteira Binária.

Alguns trabalhos já realizados com relação à escala de serviços são apresentados no Capítulo IV.

O Capítulo V mostra, através de um exemplo, como o Programa *LINGO* foi utilizado para a realização deste trabalho.

A formulação matemática para a solução do Problema de Escala de Soldados, com e sem militares de sobreaviso (para o Serviço de Plantão) é apresentada no Capítulo VI.

A implementação computacional e a avaliação dos resultados são encontradas no Capítulo VII.

Finalmente, no Capítulo VIII, encontram-se as conclusões e as sugestões para trabalhos futuros.

CAPÍTULO II

2. DESCRIÇÃO DO PROCESSO DE HIERARQUIA MILITAR E O PROBLEMA DE ESCALAS

2.1 INTRODUÇÃO

Segundo a lei número 6.880, de 09 de dezembro de 1980, que dispõe sobre o Estatuto dos Militares, "a hierarquia e a disciplina são a base institucional das Forças Armadas. A autoridade e a responsabilidade crescem com o grau hierárquico. A hierarquia militar é a ordenação da autoridade, em níveis diferentes, dentro da estrutura das Forças Armadas. A ordenação se faz por postos ou graduações; dentro de um mesmo posto ou graduação se faz pela antiguidade no posto ou graduação".

Este capítulo tem como objetivo mostrar os aspectos legais da hierarquia militar e apresentar o problema de escala de serviço de guarda dos militares.

2.2 FORÇAS ARMADAS

As Forças Armadas, essenciais à execução da política de segurança nacional, são constituídas pela Marinha, pelo Exército e pela Aeronáutica, e destinam-se a defender a Pátria e a garantir os poderes constituídos, a lei e a ordem. São instituições nacionais, permanentes e regulares, organizadas com base na hierarquia e na disciplina, sob a autoridade suprema do Presidente da República e dentro dos limites da lei (Estatuto dos Militares, Título I, Cap. I).

Organizam-se nos Comandos da Marinha, do Exército e da Aeronáutica, subordinados ao Ministro de Estado da Defesa, dispondo de estruturas e organizações próprias, definidas em legislação específica.

Ao Comando da Marinha compete: formular a política naval e a doutrina militar naval, propor a constituição, a organização e os efetivos (o número de militares de diversos graus que a compõe), bem como executar o aprestamento das forças navais; formular o planejamento estratégico e executar o emprego das Forças Navais na defesa do País; orientar e realizar estudos e pesquisas de seu interesse; contribuir para a formulação e condução de políticas nacionais que digam respeito ao mar; orientar e controlar a marinha mercante e suas atividades correlatas, no que interessa à defesa nacional; prover a segurança da navegação aquaviária e a salvaguarda da vida humana no mar; produzir material bélico de seu interesse; realizar o adestramento militar e a supervisão de adestramento civil no interesse da segurança da navegação nacional; executar a inspeção naval; e implementar e fiscalizar o cumprimento de leis e regulamentos, no mar e nas águas interiores, em coordenação com outros órgãos do Poder Executivo, Federal ou Estadual, quando se fizer necessária, em razão de competências específicas (Decreto nº 3.466, de 17 de maio de 2000, art. 31).

Ao Comando do Exército compete: formular a política e a doutrina militar terrestre; propor a constituição, a organização e os efetivos, bem como aparelhar e adestrar as forças terrestres; realizar estudos e pesquisas de seu interesse; formular o planejamento estratégico no que concerne à ação do Exército e executar ações relativas à defesa do País; participar na defesa da fronteira marítima e na defesa aérea; participar no preparo e na execução da mobilização e desmobilização nacionais; fiscalizar as atividades envolvendo armas, munições, explosivos e outros produtos de interesse militar; e produzir material bélico de seu interesse. (Decreto nº 3.466, de 17 de maio de 2000, art. 32).

Ao Comando da Aeronáutica compete: formular e conduzir a política aeronáutica nacional, civil e militar; propor a constituição, a organização e os efetivos, bem como aparelhar e adestrar a Força Aérea Brasileira; formular o planejamento estratégico e executar ações relativas à defesa do País, no campo aeroespacial; contribuir para a formulação e condução da política nacional de desenvolvimento das atividades espaciais; operar o Correio Aéreo Nacional; orientar, coordenar e controlar as atividades de aviação civil; estabelecer, equipar e operar, diretamente ou mediante concessão, a infra-estrutura aeroespacial, aeronáutica e aeroportuária; incentivar e realizar atividades de pesquisa e desenvolvimento relacionadas com as atividades aeroespaciais; estimular a indústria aeroespacial; e prover a segurança da navegação aérea. (Decreto nº 3.466, de 17 de maio de 2000, art. 33).

Os membros das Forças Armadas, em razão de sua destinação constitucional, formam uma categoria especial de servidores da Pátria e são denominados militares.

2.3 Quadros, Armas e Serviços

As Forças Armadas (como já citado na seção 2.2) são constituídas pela Marinha, pelo Exército e pela Aeronáutica.

O Exército possui elementos que, conforme sua destinação, podem ser de combate (armas-base), os quais pertencem às armas de Infantaria e Cavalaria; de apoio ao combate, constituídos pelas armas de Artilharia, Engenharia e Comunicações, e finalmente, pelos elementos de apoio logístico, isto é, os pertencentes aos serviços de Intendência e Saúde e ao quadro de Material Bélico. Eles atendem às atividades-fim do Exército, enquanto outros serviços e quadros atendem as atividades-meio. Quando agrupados esses elementos formam as unidades e subunidades de tropa: batalhões, regimentos, grupos, companhias, esquadrões e baterias.

Os quadros do Exército são: O Quadro Auxiliar de Oficiais (QAO), o Quadro Complementar de Oficiais (QCO), e o Quadro de Engenheiros Militares e os serviços do

Exército Brasileiro são o Serviço de Assistência Religiosa (SARE), e o Serviço de Saúde. (http://www.exercito.gov.br)

O Poder Naval da Marinha está estruturado, fundamentalmente, na Esquadra. As forças Distritais, compostas de navios destinados à patrulha e ao socorro e salvamento, podem apoiar determinadas ações das forças navais. A esquadra é subdividida em forças, organizadas de acordo com o meio ambiente em que suas unidades operam, que são as Forças Operativas, Centro de Apoio aos Sistemas Operativos (CASOP) a Força de Superfície Força de Submarinos, Força Aeronaval, Distritos Navais, Base Naval do Rio de Janeiro e o Arsenal de Marinha (Estaleiros de construção e reparo), Comando da Força de Minagem e Varredura e Comandos das Flotilhas de Matogrosso e Amazonas Ocidental. (Marinha do Brasil – http://www.mar.mil.br/brmar.htm)

Os Quadros do Corpo de Oficias da Ativa da Aeronáutica são os seguintes:

- Quadros de Oficiais de Carreira: Quadro de Oficiais Aviadores (QOAv); Quadro de Oficiais Engenheiros (QOEng); Quadro de Oficiais Intendentes (QOInt); Quadro de Oficiais Médicos (QOMed); Quadro de Oficiais Dentistas (QODent); Quadro de Oficiais Farmacêuticos (QOFarm); Quadro de Oficiais de Infantaria da Aeronáutica (QOInf); Quadro de Oficiais Capelães (QOCapl); Quadro de Oficiais Especialistas em Aviões (QOEAv); Quadro de Oficiais Especialistas em Comunicações (QOECom); Quadro de Oficiais Especialistas em Armamento (QOEArm); Quadro de Oficiais Especialistas em Fotografia (QOEFot); Quadro de Oficiais Especialistas em Meteorologia (QOEMet); Quadro de Oficiais Especialistas em Controle de Tráfego Aéreo (QOECTA); Quadro de Oficiais Especialistas em Suprimento Técnico (QOESup); Quadro de Oficiais Especialistas da Aeronáutica (QOEA).

- Quadros de Oficiais Temporários: Quadro Complementar de Oficiais da Aeronáutica (QCOA); Quadro de Oficiais da Reserva não Renumerada Convocados (QOCon) (Decreto no. 1145, de 20 de maio de 1994).

2.4 Círculo, Posto, Graduação e Funções

Círculos hierárquicos são âmbitos de convivência entre os militares da mesma categoria e têm a finalidade de desenvolver o espírito de camaradagem, em ambiente de estima e confiança, sem prejuízo do respeito mútuo. A seguir e no Quadro 2.1, define-se os círculos hierárquicos e a escala hierárquica nas Forças Armadas, e a correspondência entre os postos e as graduações da Marinha, Exército e Aeronáutica.

Posto é o grau hierárquico do oficial, conferido por ato do Presidente da República ou do Ministro de Força Singular e confirmado em Carta Patente. Os postos de Almirante, Marechal e Marechal-do-Ar somente serão providos em tempo de guerra.

Graduação é o grau hierárquico da praça, conferido pela autoridade militar competente. Os graduados auxiliam ou complementam as atividades dos oficiais, quer no adestramento e no emprego de meios, quer na instrução e na administração.

No exercício das atividades e no comando de elementos subordinados, os suboficiais, os subtenentes e os sargentos deverão impor-se pela lealdade, pelo exemplo e pela capacidade profissional e técnica, incumbindo-lhes assegurar a observância minuciosa e ininterrupta das ordens, das regras do serviço e das normas operativas pelas praças que lhes estiverem diretamente subordinadas e a manutenção da coesão e do moral das mesmas praças em todas as circunstâncias.

Os Guardas-Marinha, os Aspirantes-a-Oficial e os alunos de órgãos específicos de formação de militares são denominados praças especiais.

Os Cabos, Taifeiros-Mores, Soldados-de-Primeira-Classe, Taifeiros-de-Primeira-Classe, Marinheiros, Soldados, Soldados-de-Segunda-Classe e Taifeiros-de-Segunda-Classe são, essencialmente, elementos de execução.

Os Marinheiros-Recrutas, Recrutas, Soldados-Recrutas e Soldados-de-Segunda-Classe constituem os elementos incorporados às Forças Armadas para a prestação do serviço militar inicial.

Às praças especiais cabe a rigorosa observância das prescrições dos regulamentos que lhes são pertinentes, exigindo-se inteira dedicação ao estudo e ao aprendizado técnico-profissional. Às praças especiais também se assegura a prestação do serviço militar inicial. (Estatuto dos Militares, Capítulo II, Seção III).

2.5 ANTIGUIDADE E HIERARQUIA MILITAR

A precedência entre militares da ativa do mesmo grau hierárquico, ou correspondente, é assegurada pela antiguidade no posto ou graduação, salvo nos casos de precedência funcional estabelecida em lei.

A antigüidade em cada posto ou graduação é contada a partir da data da assinatura do ato da respectiva promoção, nomeação, declaração ou incorporação, salvo quando estiver taxativamente fixada outra data. Havendo empate, a antigüidade será estabelecida entre militares do mesmo Corpo, Quadro, Arma ou Serviço, pela posição nas respectivas escalas numéricas ou registros existentes em cada Força. Nos demais casos, pela antigüidade no posto ou graduação anterior, e se subsistir a igualdade, recorrer-se-á, sucessivamente, aos graus hierárquicos anteriores, à data de praça e à data de nascimento para definir a procedência (o de mais idade será considerado o mais antigo).

Na existência de mais de uma data de praça, inclusive de outra Força Singular, prevalece a antigüidade do militar que tiver maior tempo de efetivo serviço na praça anterior ou nas praças anteriores.

Entre os alunos de um mesmo órgão de formação de militares, a antiguidade será determinada de acordo com o regulamento do respectivo órgão, se não estiverem especificamente enquadrados em nenhum dos casos acima citados.

Em igualdade de posto ou de graduação, os militares da ativa têm precedência sobre os da inatividade, e entre os militares de carreira na ativa e os da reserva remunerada ou não, que estejam convocados, é definida pelo tempo de efetivo serviço no posto ou graduação.

A precedência entre as praças especiais e as demais praças é assim regulada:

I – os Guardas-Marinha e os Aspirantes-a-Oficial são hierarquicamente superiores às demais praças;

II – os Aspirantes, alunos da Escola Naval, e os Cadetes, alunos da Academia Militar das Agulhas Negras e da Academia da Força Aérea, bem como os alunos da Escola de Oficiais Especialistas da Aeronáutica, são hierarquicamente superiores aos suboficiais e aos subtenentes.

III - os alunos de Escola Preparatória de Cadetes e do Colégio Naval têm precedência sobre os Terceiros-Sargentos, aos quais são equiparados;

IV - os alunos dos órgãos de formação de oficiais da reserva, quando fardados, têm precedência sobre os Cabos, aos quais são equiparados e;

V - os Cabos têm precedência sobre os alunos das escolas ou dos centros de formação de sargentos, que a eles são equiparados, respeitada, no caso de militares, a antigüidade relativa (Estatuto dos Militares, Título I, Capítulo III).

QUADRO 2.1 – ESCALA HIERÁRQUICA NAS FORÇAS ARMADAS

Hierarquização			Marinha	Exército	Aeronáutica
Círculo de Oficiais	Círculo de Oficiais-Generais	Postos	Almirante Almirante-de-Esquadra Vice-Almirante Contra-Almirante	Marechal General-de-Exército General-de-Divisão General-de-Brigada	Marechal-do-Ar Tenente-Brigadeiro Major-Brigadeiro Brigadeiro
	Círculo de Oficiais Superiores		Capitão-de-Mar-e-Guerra Capitão-de-Fragata Capitão-de-Corveta	Coronel Tenente-Coronel Major	Coronel Tenente-Coronel Major
	Círculo de Oficiais Intermediários		Capitão-Tenente	Capitão	Capitão
	Círculo de Oficiais Subalternos		Primeiro-Tenente Segundo-Tenente	Primeiro-Tenente Segundo-Tenente	Primeiro-Tenente Segundo-Tenente
Círculo de Praças	Círculo de Suboficiais Subtenentes e Sargentos	Graduações	Suboficial Primeiro-Sargento Segundo-Sargento Terceiro-Sargento	Subtenente Primeiro-Sargento Segundo-Sargento Terceiro-Sargento	Suboficial Primeiro-Sargento Segundo-Sargento Terceiro-Sargento
	Círculo de Cabos e Soldados		Cabo	Cabo e Taifeiro-Mor	Cabo e Taifeiro-Mor
			Marinheiro Especializado e Soldado Especializado Marinheiro e Soldado Marinheiro-Recruta e Recruta	Soldado e Taifeiro de Primeira-Classe Soldado-Recruta e Taifeiro Primeira Classe	Soldado de Primeira-Classe e Taifeiro de Primeira-Classe Soldado de Segunda Classe e Taifeiro de Segunda Classe
Praças Especiais	Freqüentam o círculo de Oficiais Subalternos		Guarda-Marinha	Aspirante-a-Oficial	Aspirante-a-Oficial
	Excepcionalmente ou em reuniões sociais têm acesso aos círculos dos Oficiais		Aspirante (Aluno da Escola Naval)	Cadete (Aluno da Academia Militar)	Cadete (Aluno da Academia da Força Aérea) e Aluno da Escola de Oficiais Especialistas da Aeronáutica
			Aluno do Colégio Naval	Aluno da Escola Preparatória de Cadetes do Exército	Aluno da Escola Preparatória de Cadetes-do-Ar
			Aluno de órgão de Formação de Oficiais da Reserva	Aluno de órgão de Formação de Oficiais da Reserva	Aluno de órgão de Formação de Oficiais da Reserva
	Excepcionalmente ou em reuniões sociais têm acesso ao círculo de Suboficiais, Subtenentes e Sargentos		Aluno de Escola ou Centro de Formação de Sargentos	Aluno de Escola ou Centro de Formação de Sargentos	Aluno de Escola ou Centro de Formação de Sargentos
	Freqüentam o círculo de Cabos e Soldados		Aprendiz-Marinheiro Aluno de órgão de Formação de Praças da Reserva	Aluno de órgão de Formação de Praças da Reserva	

2.6 O Problema de Escalas

Todos os militares têm um tipo de serviço chamado de administrativo. Neste serviço eles normalmente obedecem a um horário comum como qualquer outro trabalhador. Por exemplo, eles podem trabalhar de segunda a sexta-feira das 8:00 às 18:00h.

No entanto eles devem também cumprir um outro tipo de horário, chamado horário de guarda, onde trabalham 24 horas sem descanso. Normalmente esta escala especial de serviço inicia e termina às 9:00 horas da manhã.

Atualmente esta escala é feita manualmente, o que acarreta um trabalho que leva até 2 dias para ser concluído. Mesmo considerando a hierarquia, a grande maioria dos militares fica insatisfeita com a escala de serviço de guarda. Se um militar falta, não existe critério para a sua substituição. Em geral, quem já está no plantão permanece por mais um dia para cobrir a falta. Cada mês é tratado isoladamente.

O objetivo deste trabalho é determinar uma escala ótima de serviço de guarda, obedecendo a certos critérios que serão posteriormente discutidos, procurando aumentar a satisfação dos militares e facilitar o trabalho do responsável pelas escalas de plantão.

O militar (no presente trabalho, trata-se de Soldado de Primeira-Classe e Soldado de Segunda-Classe da Aeronáutica, cujo enquadramento pode ser observado no quadro 2.1 anterior), não deverá prestar serviço administrativo no dia em que terminou um cumprimento de guarda. Isto significa que o militar terá folga de qualquer serviço neste dia.

As características básicas que devem ser conhecidas sobre este serviço estão listadas a seguir:

- tempo de trabalho por folga mínimo (24/24 – 24/48 – 24/72 – 24/96...), ou seja, o militar cumpre guarda um dia e folga um dia (24/24), cumpre guarda um dia e folga dois dias (24/48) e assim por diante. Porém, para um bom desenvolvimento de serviço do militar, é de interesse que o cumprimento de guarda seja no mínimo na escala 24/72, ou seja, se ele cumpre guarda um dia, não cumprirá guarda no mínimo nos três dias seguintes;

- não cumprir guarda em finais de semana seguidos. É desejável que o militar que cumpre guarda num sábado ou num domingo de uma semana, não o faça no final de semana seguinte;
- não trabalhar mais que um número determinado de finais de semana por mês. Considerando um mês que tenha 5 finais de semana, é limitado a 3 finais de semana no máximo o cumprimento de guarda. Num mês de 4 finais de semana, não mais que dois finais de semana serão de guarda para o militar;
- não exceder a uma carga mensal de dias de guarda para cada militar. Cada militar terá um determinado número fixo de dias em que deverá cumprir guarda num mês, de acordo com o número de militares disponíveis para trabalhar;
- atender a uma determinada demanda diária de militares para serviço de guarda. Normalmente essa demanda diária é fixa. Apenas em casos especiais, por exemplo, no caso da presença de uma autoridade, o número de militares de guarda pode ser maior;
- possibilitar a escolha de dias de trabalho pelos militares obedecendo a critérios de hierarquia, dentro do possível. Essa escolha poderá acarretar em uma melhor utilização dos feriados e finais de semana, devido à folga que o militar ganha no dia em que deixa a guarda, e também para viabilizar seus compromissos particulares.

No próximo capítulo pretende-se abordar alguns problemas e métodos exatos para solução de problemas de Programação Linear Inteira Binária (PLIB), pois o problema de escala de militares é resolvido neste trabalho como um PLIB.

CAPÍTULO III

3. DESCRIÇÃO DE MÉTODOS EXATOS PARA A SOLUÇÃO DOS PROBLEMAS BINÁRIOS

3.1 INTRODUÇÃO

Aproveitando a característica hierarquizada da estrutura militar, o problema de escala de serviço de plantão de militares pode ser modelado atribuindo-se pesos a cada um deles e usando uma estrutura binária para as variáveis de decisão (trabalha ou não trabalha em um determinado dia), tem-se que o mesmo pode se modelado como um problema de Programação Linear Inteira Binária. Neste capítulo, pretende-se mostrar uma variedade de problemas semelhantes a este aqui abordado, com o objetivo de indicar possíveis caminhos para a obtenção da solução.

A Programação Linear (PL) é um meio matemático de designar um montante fixo de recursos que satisfaça certa demanda de tal modo que alguma função objetivo seja otimizada e ainda satisfaça a outras condições definidas (Shamblin, 1989).

As variáveis na PL são contínuas e apresentam comportamento linear, tanto em relação às restrições como à função objetivo.

Quando qualquer variável não puder assumir valores contínuos, mas apenas inteiros, o modelo de otimização passa a ser um problema de Programação Linear Inteira (PLI), que será tema deste capítulo. Uma variante é a Programação Inteira Mista (PLIM), em que algumas variáveis são inteiras e outras são contínuas.

Um outro modelo também importante é aquele em que as variáveis do problema só podem assumir valores 0 ou 1. Neste caso o modelo é denominado de problema de

Programação Linear Inteira 0-1 ou problema de Programação Linear Inteira Binária, que também será abordado neste capítulo.

3.2 MÉTODOS EXATOS PARA A SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO INTEIRA E BINÁRIA

Podem-se considerar dois métodos de solução de problemas: os métodos exatos, que apresentam uma solução exata e os métodos heurísticos, que na maioria das vezes não apresentam uma solução ótima, mas quase ótima.

Neste trabalho serão apresentados alguns métodos exatos para solução de problemas de programação inteira.

3.2.1 Método *Branch-and-Bound*

Um método simples para resolver problemas de Programação Inteira e Inteira Mista foi publicado em 1960, em um artigo de A. H. Land e A. G. Doig. Esse método, *Branch-and-bound*, efetua partições no espaço das soluções (*branch*), e prova a otimalidade da solução usando limites calculados ao longo da enumeração dos pontos candidatos à solução (*bound*).

3.2.1.1 Descrição do Algoritmo

Considerando um problema de maximização da função objetivo de um problema de Programação Linear Inteira Pura (PLIP), os passos para o desenvolvimento do algoritmo são (Land and Doig, 1960):

Passo 1. calcular a solução ótima, relaxando a condição de integralidade. Se a solução contínua é factível para o PLIP, então a solução é ótima, parar. Caso contrário, estabelecer um limite inferior finito para o valor da função objetivo, igual ao valor desta num ponto factível para o PLIP. Se a solução contínua não é factível para o

PLIP e não é possível estabelecer um limite inferior finito para o valor da função objetivo, considere-se o limite inferior igual a $(-\infty)$;

Passo 2. partição (*branch*). Selecionar um dos subespaços de solução ainda não explorados (na 1ª. iteração é considerado todo o espaço de solução do problema original relaxando a condição de integralidade) e em que a solução contínua não é factível; nesta solução selecionar uma variável não inteira para a partição do subespaço em apreciação. Admitindo que esta variável tem valor não inteiro α, estabelecem-se duas novas restrições em que a variável é respectivamente $\leq$ e $\geq$ aos valores inteiros adjacentes de α. Aumentar o problema com cada uma destas novas restrições constituindo assim dois novos subproblemas (descendentes). Enumerar através de uma árvore possibilidades de solução dos problemas gerados pela divisão do problema;

Passo 3. limite (*bound*). Calcular a solução ótima destes subproblemas relaxando a condição de integralidade. O valor da função objetivo constitui o valor máximo em cada subespaço;

Passo 4. avaliação. Eliminar de futura análise toda solução não factível em ambiente contínuo (não pode gerar soluções factíveis inteiras); toda solução com valor da função menor ou igual ao limite inferior atual, independentemente de ser ou não factível (não poderá gerar soluções com melhor valor para a função); toda solução que é factível para o PLIP e tem valor superior ao limite inferior atual, passa a constituir novo limite inferior;

repetir o Passo 2 se ainda há problemas para partição;

repetir os Passos 3 e 4 se ainda há soluções para avaliar;

parar.

Um dos pontos fundamentais para o sucesso do *Branch-and-Bound* é a qualidade do limite gerado pela solução inteira. Essa qualidade normalmente depende da estratégia de desdobramento da árvore de busca. Existem basicamente duas grandes estratégias, o da busca em profundidade, onde, estabelecida uma partição escolhe-se um dos subproblemas descendentes para resolver e neste, se há partição volta a escolher-se um dos descendentes para resolver e repete-se o procedimento até não ser necessária nova partição e então se regressa ao nível imediatamente anterior onde há partição não avaliada e se repete o procedimento. E os da busca em largura são estabelecidos e estudados os subproblemas do mesmo nível antes de efetuar partições e passar ao nível seguinte. No mesmo nível estudam-se em primeiro lugar os subproblemas descendentes de problemas onde a função tem maior valor (procurando obter um limite inferior tão elevado quanto possível).

Para a escolha da variável de partição, Dakin (1965) propõe que seja a que possuir maior resíduo em relação à solução inteira. Por exemplo, se $x_1 = 8{,}35$ e $x_2 = 12{,}85$, a variável a ser escolhida será x_1, pois tem resíduo igual a 0,35, sendo que para x_2 tem-se um resíduo igual a 0,15.

Land e Doig (1960), sugerem que a árvore seja expandida em vários valores simultâneos para a variável de divisão. A variante de Spielberg (1968) [Goldbarg, 2000] desenvolve o nó com maior valor de z^* e o mais recentemente calculado, usando o critério de Land e Doig para aumentar a retirada de espaço contínuo no entorno do nó pesquisado. Trata-se de uma busca em profundidade associada ao critério de Land e Doig.

Outras técnicas complementares para obtenção dos limites são a Relaxação Linear, Relaxação Lagrangeana, Algoritmos Heurísticos e Cortes.

3.2.2 Programação Linear Inteira Binária

A formulação de problemas com variáveis de decisão binárias permite incorporar decisões do tipo SIM/NÃO ao modelo. Essas variáveis são restritas a dois valores, $x_j = 1$ se a decisão for sim e $x_j = 0$ caso contrário, e também podem permitir reformular um problema que, devido a não convexidade, não pode ser administrado como um problema de Programação Linear (PL).

3.2.2.1 Problemas de Programação Linear Inteira Binária

Um importante modelo de Programação Linear Inteira é o problema da mochila, pois pode ser aplicado em vários contextos e existem vários algoritmos e aproximações para a sua resolução. Resumidamente, um Problema da Mochila consiste na escolha de um subconjunto de itens, cada qual com uma correspondente utilidade e um valor (em geral denominado "peso") que define o quanto esse item utilizará da capacidade da mochila.

Pode-se formular o problema da seguinte forma:

$$Max\ z = \sum_{j=1}^{n} c_j x_j$$

$$s.a.$$

$$\sum_{j=1}^{n} w_j x_j \leq b$$

$$x_j \geq 0 \text{ e inteiros}$$

onde a variável x_j representa o número de itens do tipo j selecionados; os valores de c_j representam os valores econômicos (as utilidades) de cada item, e w_j os pesos dos itens correspondentes e b o peso total.

Diferentes tipos de problemas de mochila podem aparecer, por exemplo, considerando que exista apenas um item de cada tipo para ser escolhido, a restrição da variável inteira é substituída por $x_j \in \{0,1\}$, e tem-se o Problema da Mochila 0-1, também denominado de problema da mochila unidimensional.

Sua formulação pode ser da seguinte maneira:

$$Max\ z = \sum_{j=1}^{n} c_j x_j$$

$$s.a. \begin{cases} \sum_{j=1}^{n} w_j x_j \leq b \\ x_j \in \{0,1\} \quad j = 1,\ldots,n \end{cases}$$

Podem-se citar outras variantes deste problema, como o Problema da Mochila com Limites; o Problema da soma de subconjuntos, também denominado por Christofides et al. (1979) por *Value-Independent Knapsack Problem* (Problema da Mochila de Valor Independente); o Problema Múltiplo 0-1; o problema da Mochila 0-1 Multidimensional; o Problema Quadrático da Mochila (Gallo et al., 1980); o problema da Mochila Max-Min 0-1; o problema da Mochila de Escolha Múltipla (Dudzinski e Walukiewicz, 1987); o problema da Mochila Encapsulada.

Uma das técnicas para resolução de Problemas da Mochila é a Programação Dinâmica que será abordada a seguir.

3.2.3 Programação Dinâmica

O emprego da Programação Dinâmica (PD) foi intensamente estudado em Problemas da Mochila. A PD é uma técnica utilizada para a otimização de processos de decisão multiestágios que podem ser desdobrados segundo um certo número de etapas seqüenciais ou estágios. As alternativas incluídas na conclusão de um estágio são denominadas decisões. A condição do processo dentro de cada estágio é denominada estado. Cada estágio inclui a tomada de uma decisão que pode ou não alterar o estado do processo, mas que, obrigatoriamente, representa uma transição entre o estado corrente e o estado futuro do processo. Esse processo é finito quando existe apenas um número finito de estágios no processo e um número finito de estados possíveis associado a cada estágio. Dentro do processo multiestágios, o objetivo do tomador de decisão é encontrar uma política ótima (trajetória ótima) em relação ao retorno auferido com as decisões. Um processo de decisão multiestágios é determinado se o resultado de cada decisão for conhecido exatamente (Goldbarg, 2000).

Num sistema de Programação Dinâmica vale o princípio da otimalidade: "Para um dado sistema, a política ótima para os estados remanescentes é independente da política de decisão adotada em estados anteriores" (Bellman, 1957).

Para garantir a otimalidade da decisão inerente à mudança de um estado de uma dada etapa para outro estado da etapa seguinte, é necessário estabelecer uma relação matemática que defina o encargo associado à transição (função transição).

Os problemas de PD podem ser resolvidos recuando temporalmente (*backward*) no momento da decisão, iniciando-se o processo de cálculo na última etapa do problema (última decisão) e deduzindo a partir desta, sucessivamente, as decisões ótimas das etapas antecessoras até atingir a etapa inicial.

Para a resolução de problemas como o da mochila, ou qualquer outro que apresente somente variáveis binárias um método bastante conhecido é o Algoritmo de Balas, que será abordado a seguir.

3.2.4 Algoritmo de Balas

O Algoritmo de Balas (E. Balas, 1965), consiste em igualar a zero tantas variáveis quanto permitam as restrições para dar preferência ao uso das variáveis que tem subíndices menores quando necessário igualar a 1.

3.2.4.1 Introdução

Considerando a forma padrão para o Problema de Programação Linear Binária (PPLB) abaixo (Mayerle, 2001):

$$\begin{aligned} &Max \quad z = \sum_{j=1}^{n} c_j x_j \\ &s.a. \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1,\dots,n \\ &\qquad\quad x_j \in \{0,1\} \qquad\quad j = 1,\dots,n \end{aligned}$$

no qual $c_1 \geq c_2 \geq \dots \geq c_n \geq 0$, pode acontecer de um problema não se adaptar à esta forma, devendo então sofrer algumas transformações:

a) efetuar a substituição $x_j = 1 - x_j^{'} \ \forall c_j < 0$;

b) trocar cada uma das restrições do tipo $\sum a_{ij} x_j = b_i$ por duas restrições: uma do tipo $\sum a_{ij} x_j \leq b_i$ e outra do tipo $\sum a_{ij} x_j \geq b_i$;

c) trocar cada uma das restrições do tipo $\sum a_{ij} x_j \geq b_i$ por uma restrição $-\sum a_{ij} x_j \leq -b_i$;

d) reordenar as variáveis do PPLB de modo que $c_1 \geq c_2 \geq \dots \geq c_n$.

O algoritmo consiste em um processo de busca em uma árvore binária, na qual cada nó n da árvore corresponde a uma solução parcial do PPLB, cujos valores das k primeiras variáveis são conhecidas. Para verificar a possibilidade de existência de uma solução viável e ótima a partir desta solução parcial, devem-se aplicar dois testes: o teste de otimalidade e o teste de viabilidade.

O Teste de Otimalidade considera o conhecimento prévio de uma solução ótima temporária, cujo valor da função objetivo é $z_{ótimo}$. Considerando que na solução parcial as k primeiras variáveis são conhecidas, uma estimativa do máximo do valor da função objetivo para uma solução completa derivada desta solução parcial, poderá ser feita através de:

$$z_{estimado} = \sum_{j=1}^{k} c_j x_j + \sum_{j=k+1}^{n} c_j$$

Se o valor estimado de z for superior ao valor ótimo, então é possível que se encontre uma solução ótima a partir desta solução parcial. Em caso contrário, a melhor solução possível de ser encontrada não será melhor que a solução ótima temporária já disponível.

O Teste de Viabilidade considera que para cada restrição deve ser satisfeita a seguinte condição:

$$\sum_{j=1}^{k} a_{ij} x_j + \sum_{j=k+1}^{n} a_{ij} x_j \leq b_i$$

ou

$$\sum_{j=1}^{k} a_{ij} x_j \leq b_i - \sum_{j=k+1}^{n} a_{ij} x_j \leq b_i - \sum_{j=k+1}^{n} \min\left(0, a_{ij}\right)$$

Então, se para alguma restrição:

$$\sum_{j=1}^{k} a_{ij} x_j > b_i - \sum_{j=k+1}^{n} \min\left(0, a_{ij}\right)$$

conclui-se que não existirão valores para as variáveis ainda não conhecidas que permitam a obtenção de uma solução que satisfaça a restrição.

No caso de falha de um destes dois testes, é necessário rever os valores atribuídos às k primeiras variáveis.

3.2.4.2 Descrição do Algoritmo de Balas

Passo 0. Construir o PPLB em sua forma padrão. Fazer $z_{ótimo} = -\infty$ e $k = 1$;

Passo 1. fazer $x_k = 1$ e verificar se:

$$z_{estimado} = \sum_{j=1}^{k} c_j x_j + \sum_{j=k+1}^{n} c_j > z_{ótimo}$$

e

$$\sum_{j=1}^{k} a_{ij} x_j \leq b_i - \sum_{j=k+1}^{n} \min(0, a_{ij}) \quad i = 1,\ldots,m.$$

Em caso de sucesso, ir ao passo 3;

Passo 2. fazer $x_k = 0$ e verificar se:

$$z_{estimado} = \sum_{j=1}^{k} c_j x_j + \sum_{j=k+1}^{n} c_j > z_{ótimo}$$

e

$$\sum_{j=1}^{k} a_{ij} x_j \leq b_i - \sum_{j=k+1}^{n} \min(0, a_{ij}) \quad i = 1,\ldots,m.$$

Em caso de fracasso, ir ao passo 4;

Passo 3. fazer $k = k + 1$. Se $k \leq n$, volte ao passo 1. Senão, uma solução completa, melhor que a solução ótima temporária foi encontrada. Guardar esta solução como sendo a nova solução ótima temporária. Atualizar o valor de $z_{ótimo}$;

Passo 4. *backtrack*. Encontrar $K = \{j / x_j = 1 \text{ e } 0 < j < k\}$. Se $K = \emptyset$, então parar. A solução ótima temporária é a solução ótima do problema. Em caso contrário, determinar $k = \max\{j / j \in K\}$ e retornar ao passo 2.

Um outro importante algoritmo desenvolvido para a Programação Linear Inteira é o *Branch-and-Cut*, abordado a seguir, introduzido por Padberg e Rinaldi em 1991, no contexto do Problema do Caixeiro Viajante. Hoffman e Padberg, também em 1991, aplicaram o algoritmo para programas puros 0-1.

3.2.5 Algoritmo *Branch-and-Cut*

O método *Branch-and-Cut* consiste de uma combinação de um método de corte plano com um algoritmo *Branch-and-Bound*. Este método trabalha resolvendo uma seqüência de relaxações da Programação Linear de Problemas de Programação Linear Inteira (Mitchell, 1991). A geração automática de cortes é executada não somente antes do começo do *Branch-and-Bound* como também para cada nó da árvore enumerada, e deve ser válido para todos os nós (Balas et al., 1996).

Considerando o problema de Programação Linear Mista com Variáveis Binárias (PLMB)

$$Min\ cx$$

$$s.a. \begin{cases} Mx \geq d \\ x \geq 0 \\ x_i \in (0,1) \quad i = 1,\dots,p \end{cases}$$

onde as primeiras p variáveis são restrições 0-1 e as variáveis restantes $x_i, i = p+1,\dots,n$ são contínuas.

Para o passo genérico do algoritmo *Branch-and-Cut*, a Relaxação da Programação Linear original, $Mx \geq d, x \geq 0, x_i \leq 1, i = 1,\dots,p$ é enriquecida pela adição de inequações válidas para PLMB e algumas das variáveis restritas 0-1 são fixadas ou acima ou abaixo do limite. Considerando C a família de inequações válidas para PLMB, e assumindo que o sistema linear $Ax \geq b$ definido em C contém pelo menos todas as inequações em

$Mx \geq d, x \geq 0, x_i \leq 1, i = 1,...,p$. Denotando os conjuntos de variáveis fixadas em 0 e 1 por $F_0, F_1 \subseteq \{1,...,p\}$ respectivamente, e dado:

$$K(C, F_0, F_1) = \{x / Ax \geq b, x_i = 0 \text{ para } i \in F_0 \text{ e } x_i = 1 \text{ para } i \in F_1 \}$$

e considerando que $PL(C, F_0, F_1)$ denota o programa linear

$$Min\ cx$$
$$x \in K(C, F_0, F_1)$$

assumido factível, com um mínimo finito. Representando os nós ativos da árvore de enumeração por uma lista δ de pares ordenados (f_0, F_1) e considerando UB a representação do limite atual superior, ou seja, o melhor valor conhecido para o PLMB, tem-se o algoritmo a seguir.

3.2.5.1 Descrição do Algoritmo *Branch-and-Cut*

(Entrada *c, M, d, p*)

Passo 1. Inicialização. Conjunto $\delta = \{(F_0 = \varnothing, F_1 = \varnothing)\}$, sendo C a relaxação do PL do PLMB e $UB = \infty$.

Passo 2. Seleção do nó. Se $\delta = \varnothing$, parar. Caso contrário, escolher um par ordenado $(f_0, F_1) \in \delta$ e tirar do δ.

Passo 3. Limite superior. Resolver o programa linear $LP(C, F_0, F_1)$. Se o problema é infactível, ir ao passo 2. Se $\overline{x}_j \in \{0,1\}$, j = 1,...,p, tomar $x^* = \overline{x}$, $UB = c\overline{x}$ e ir ao passo 2.

Passo 4. Decisão - limitar ou cortar. Planos de corte podem ser gerados? Se sim, ir ao passo 5 e após ao passo 6.

Passo 5. Geração de corte. Produzir planos de corte $\alpha\ x \geq \beta$ válidos para PLMB mas violado por $\bar{x}$. Adicionar os cortes em C e ir ao passo 3.

Passo 6. *Branching*. Escolher um índice $j \in \{1,\ldots,p\}$ tal que $0 < \bar{x}_j < 1$. Gerar subproblemas correspondentes em $(F_0 \cup \{j\}, F_1)$ e $(F_0, F_1 \cup \{j\})$, calcular seus limites superiores e adicionar ao δ. Ir ao passo 2.

Quando o algoritmo para, se $UB < \infty$, então x^* é uma solução ótima para o PLMB, caso contrário, o problema é infactível.

A seguir, é apresentado o uso de planos de corte *Lift-and-Project* no algoritmo *Branch-and-Cut*.

3.2.6 Corte *Lift-and-Project* em um Algoritmo *Branch-and-Cut* para Problemas de Programação Inteira Mista

A combinação da geração de planos de cortes com *Branch-and-Bound* resulta num procedimento versátil para resolver problemas inteiros mistos.

Balas et al. em 1996 investigaram o uso de planos de corte no passo 5 do algoritmo *Branch-and-Cut* utilizando os cortes *Lift-and-Project*.

3.2.6.1 Cortes *Lift-and-Project*

Estes planos de cortes são assim chamados, porque eles podem ser obtidos por um procedimento que eleva (*lift*) o conjunto de restrições da relaxação do PL em um espaço dimensional mais alto, introduzindo novas variáveis, somando algumas equações com restrição 0-1 e, finalmente, projetando de volta no espaço original para eliminar as novas variáveis. A conseqüência deste processo de *lift-strengthening-projecting* é um conjunto de restrições mais fechado que o original, mas ainda satisfazendo todas as soluções factíveis do problema PLMB.

Em particular, os cortes *lift-and-project* podem ser obtidos impondo a condição 0-1 em uma única variável x_j. Estes cortes são inequações válidas para:

$$P_j(K) = conv(K \cap \{x \in \Re^n / x_i \in (0,1)\}),$$

onde $K = K(C, F_0, F_1)$.

3.2.6.2 Conclusões sobre *Lift-and-Project* em um Algoritmo *Branch-and-cut*

Após muitos testes de estratégias de planos de cortes, Balas et al (1996) concluíram que: calculando-se a distância Euclidiana entre o hiperplano definido pelo corte e o ponto fora dele, consegue-se uma medida útil de qualidade; é melhor gerar cortes em grandes *rounds* do que um de cada vez em pequenos *rounds*. Dada uma solução $\bar{x}$ da $PL(C, F_0, F_1)$ que não é factível para o PLMB, considera-se um *round* de cortes o conjunto de planos de corte gerados para algum ou todos $j \in \{1, ..., p\}$ tal que $0 < \bar{x}_j < 1$; o *Lift-and-Cut* é um meio eficiente para incorporar cortes genéricos em um *Branch-and-Cut*; a decisão de ramificar ou cortar, é uma das mais importantes buscas para resolver um *Branch-and-Cut*. Nem sempre, diminuir a árvore de enumeração significa diminuir o tempo computacional. O importante é haver um equilíbrio entre cortar e ramificar, e uma estratégia simples baseada na qualidade dos cortes para o nó de origem, mostrou-se satisfatória em vários exemplos executados por Balas et al.

O algoritmo *Branch-and-Cut* com implementação criado por Balas et al (1996) é eficiente para resolver problemas de Programação Mista com Variáveis Binárias. Em muitos exemplos de problemas de Programação Linear Inteira Pura ou Mista 0-1, obteve-se resultados tão bons, e às vezes melhores que os algoritmos disponíveis atualmente.

A principal vantagem do *Lift-and-Cut* em um *Branch-and-Bound* usado por Balas et al. (1996), é que ele é capaz de gerar planos de corte independentemente da estrutura do problema, resultando em soluções mais robustas para resolver Problemas de Programação Inteira Mista.

No próximo capítulo serão apresentadas algumas soluções para problemas de escalas de trabalho de funcionários.

CAPÍTULO IV

4. ALGUMAS SOLUÇÕES PARA PROBLEMAS DE ESCALAS DE TRABALHO

4.1 INTRODUÇÃO

Muitos foram os modelos para formação de horários de serviços desenvolvidos ou adaptados para resolver problemas específicos. Pode-se citar Barboza (2000) que propõe uma solução para a elaboração e designação de horários de atendentes em uma central telefônica de atendimento de usuários construindo um modelo de Programação Linear e utilizando o algoritmo do *Matching* de Peso Máximo para resolver o problema de designação; Siqueira (1999), que mostra a aplicação do algoritmo do *Matching* de Peso Máximo na elaboração de jornadas de trabalho para motoristas e cobradores de ônibus; Mason e Nielsen (1999) que desenvolveram um sistema para resolver automaticamente problemas de escalas. Esse sistema utiliza técnicas de otimização por restrição, em particular, uma formulação generalizada de particionamento do conjunto de escala e resolve usando Programação Linear e métodos *Branch-and-Bound*; Popova e Morton (1998) que propõem um modelo de programação para produção de escala de funcionários combinando técnicas de estatísticas bayesianas, da programação estocástica e da simulação; Constantino (1997) que apresenta um modelo de geração de escala cíclica para a aplicação em uma empresa de transporte ferroviário de carga; Kusumoto (1996) que desenvolveu um Sistema de Programação para horários de Enfermeiras que usa bibliotecas do *ILOG SOLVER* 3.0 e roda no Windows e Windows NT; Lau and Lua (1997) propõem uma abordagem de Programação por Restrição para solução de problemas de designação de tripulação, Lazaro and Aristondo (1995) entre outros.

O problema de planejamento de escala de serviços para trabalhadores (*manpower scheduling*), pode ser dividido em dois subproblemas, muitas vezes trabalhados separadamente: *scheduling problem* e *rostering problem*. Neste capítulo pretende-se mostrar alguns trabalhos já realizados para resolução de problemas de escala de pessoal.

4.2 ESCALAS DE TRABALHO

Planejamento de escala para operadores de telefone, designação de horários de tripulações em empresas aéreas, horários de enfermeiros em hospitais, horários de motoristas em empresas de ônibus, horário de telefonistas em centrais de atendimento são alguns dos problemas reais que se apresentam e são abordados em vários estudos.

Scheduling Problem e *Rostering Problem* são subdivisões do estudo de problemas de escalas de trabalho referente ao planejamento de escala de serviço e a subseqüente designação dos trabalhadores de acordo com suas preferências. A seguir, faz-se um resumo de alguns dos estudos referentes à resolução de problemas de escala de serviços, onde são consideradas as hierarquias e procura-se atender as necessidades dos funcionários e algumas condições de escala que tem características semelhantes à escala de plantão dos militares.

4.2.1 *Nurse Rostering at the Hospital Authority of Hong Kong* (Designação de Enfermeiros para o Hospital *Authority* de Hong Kong)

Chun et al. (2000) descrevem o *Rostering Engine* (RE) desenvolvido para o Hospital Authority (HA) como parte de seu *Staff Rostering System* (*SRS*, Sistema *Rostering* da equipe de funcionários) usando técnicas de programação com Inteligência Artificial.

Muitas das restrições usadas em *SRS* são únicas para o HA, mas as metas e os objetivos do *SRS* são similares para muitos outros sistemas de *rostering*. Por exemplo, o *SRS* assegura a existência de um número adequado de pessoal qualificado para manter a qualidade de serviço, sem sobrecarregar um funcionário ou não utilizar a mão de obra disponível; procura também que cada equipe tenha o mesmo número de folgas em finais de semana e feriados, ou o mesmo número de noites trabalhadas; procura considerar as necessidades de cada pessoa e fundamentalmente, tenta maximizar o intervalo entre dois serviços noturnos e prevenir ou evitar padrões de troca que não são desejáveis, por exemplo, trabalhar duas noites em domingos sucessivos ou trabalhar numa tarde e fazer uma troca noturna.

O *SRS* é um sistema computacional desenvolvido para melhorar o atendimento médico. Seu funcionamento inclui geração de escalas usando programação por restrições, impressão das escalas para distribuição para as equipes, armazenamento de registros em base de dados e geração de relatórios.

Utiliza um programa feito no *Microsoft Visual Basic* para interface entre o processamento e o usuário e para o armazenamento de informações utiliza o banco de dados *Microsoft SQL Server*.

Quando uma pessoa utiliza o *SRS*, ela deve primeiramente atualizar o sistema com mudança de dia de serviço, ou alguma promoção ou demissão ou outra alteração. Também poderá acrescentar alguma restrição ou parâmetro que possa ter modificado. Após todas as mudanças informadas, o usuário chama o *Rostering Engine* (Mecanismo de Escala) para produzir uma listagem para um particular conjunto de funcionários, trocas e períodos de *rostering*. O *Rostering Engine* cria quatro tipos principais de informação:

- informação estática sobre a enfermaria do hospital. Inclui o departamento a que esta enfermaria pertence, as trocas dentro de uma enfermaria, os serviços especiais, e como devem ser geradas as listas;

- informação de pessoal. Inclui informação em grau, antiguidade, enfermarias atribuídas, balanços e pedidos de pessoal;
- restrições e parâmetros. São as restrições e parâmetros usados durante o *rostering*;
- histórico. Inclui um conjunto de estatísticas, tais como o número de licenças acumuladas, e um histórico prévio da lista.

Guiado pelas restrições pela relação do período anterior e pedidos dos funcionários, o *Rostering Engine* produz a lista desejada para o período utilizando técnicas de programação de Inteligência Artificial (Chun, 1999, Puget 1994).

Muitos benefícios são citados no artigo, como o aumento de produtividade, maior satisfação dos funcionários, qualidade e facilidade na administração, qualidade do serviço melhorada.

4.2.2 *Operations Research in Industry – The Alitalia Experience* (Pesquisa Operacional em Indústria – A Experiência *Alitalia*)

Este trabalho, realizado por Broggio e Paoletti (2001) descreve os resultados da aplicação de um otimizador de Programação Linear Inteira Mista (PLIM) para resolver um problema de escala de equipe de terra. Os dados foram obtidos no Aeroporto Fiumicino, Divisão Alitalia.

O *Roster Planning* (plano de designação) foi analisado como um problema de Programação Inteira. Para maximizar a satisfação da equipe e otimizar a distribuição de pessoal, foram relacionadas muitas restrições.

Neste caso, a escala é composta de uma tabela de 7 colunas (os dias da semana) e muitas linhas, cujos elementos representam turnos. As linhas na escala descrevem semanas consecutivas e são distinguidas pelos números de linha/semana. O conteúdo de uma célula (intersecção entre linha e coluna) na escala é construído em ordem para garantir a mesma carga de trabalho para todos os empregados. A rotação de empregados nas linhas implica que a escala deve ser cíclica, ou seja, todas as seqüências de turnos obtidas devem ser factíveis com as leis de trabalho. A factibilidade é necessária também para a seqüência obtida ligando a última e a primeira linha. A escala pode também ser vista como a escala de turnos para *n* semanas para um dado empregado. As variáveis binárias de designação são as variáveis principais neste problema.

As restrições consideradas são as de: designação – para cada membro da equipe e para cada dia, exatamente um turno deve ser designado; cobertura de turno de trabalho – o número de membros designados deve ser maior ou igual à demanda; seqüências proibidas – para turnos da noite; turnos de descanso semanal – começando na segunda e terminando no domingo; intervalo de descanso de turno e de descanso de turno do domingo.

Pela impossibilidade da resolução usando um modelo PLIM, o *Institute of System Analysis and Computer Science of the Italian National Research Council (IASI,* Instituto de Análise de Sistema e Ciência da Computação do Conselho Italiano de Pesquisa Nacional*)* experimentou reduzir o tempo de solução aplicando a teoria poliédrica e o algoritmo *Branch-and-Cut* enquanto o *Operations Research Department of Alitalia* (*ROR* – Departamento de Pesquisa Operacional de *Alitalia*) experimentou reduzir as dimensões do problema, com a idéia de resolver partes da escala separadamente. O sucesso foi alcançado em ambas as experiências.

ROR (1999) projetou uma heurística de pesquisa local que aplicou a formulação *IASI* para subproblemas e encontrou uma boa solução global. A aproximação combinada teve a importante característica do tempo de solução ser quase independente da dimensão da escala, e pode ser expressa em termos de número de linhas da escala por hora de elaboração.

O projeto de planejamento da escala de pessoal de terra representa um sucesso de técnicas de Pesquisa Operacional e ferramentas de PLIM para resolver problemas reais.

4.2.3 *Optimal Shift Scheduling with a Global Service Level Constraint* (Escala Ótima de Equipe de Funcionários com um Nível de Restrição Global)

Koole e Sluis (1998) propõem um método que combina a determinação de nível de funcionários e a determinação de escala de turnos para centros de chamada, considerando intervalos de muito e pouco serviço. São considerados também, intervalos de mesmo tamanho, excluindo as paradas. O período de funcionamento da central toma apenas parte do dia.

Normalmente, utiliza-se a pesquisa local para resolver este tipo de problema, mas que não garante a solução ótima global. Neste caso, foi introduzido um método em que a pesquisa local converge para o ótimo global. A principal propriedade abordada foi a multimodularidade, introduzida por Hajek (1985).

Existe uma definição que relaciona uma função $f:\mathbb{Z}^m \to \mathbb{R}$ com uma função $\tilde{f}:\mathbb{R}^m \to \mathbb{R}$. Diz-se que f é multimodular se $\tilde{f}$ é convexa. A função $\tilde{f}$ é construída pelos próprios pontos inteiros da função f, acrescidos de combinações convexas de uma vizinhança de cada ponto x (denominada átomo). Assim pode-se mostrar que ao encontrar um mínimo local de uma função multimodular encontra-se seu mínimo global.

Sob certas condições, resolver um problema de escala de força de trabalho é equivalente a minimizar uma função multimodular, que fornece um método baseado em pesquisa local para encontrar a escala ótima.

Os resultados computacionais mostraram que a maioria das melhorias foi obtida de turnos simples. Isto fornece uma boa heurística para problemas de programação mais complexos que envolvem "quebras" de turnos de diferentes tamanhos. Por não ser o algoritmo exato (busca local), poderia ser uma idéia boa para implementar um método como o *simulated annealing*.

Após esta revisão retorna-se ao problema original para resolução de escala de militares. Como a formulação é feita por um PLIB e para sua solução o programa *LINGO* foi utilizado, faz-se a seguir uma análise de como utilizar este software.

CAPÍTULO V

5. UTILIZAÇÃO DO PROGRAMA *LINGO*

5.1 O QUE É O PROGRAMA

O *LINGO* é um software desenvolvido por *LINDO Sistem's Inc.* utilizado para resolver problemas de otimização envolvendo problemas de Programação Linear, Programação Linear Inteira, Programação Linear Inteira Binária, Programação Não-Linear, entre outros.

Este programa reduz drasticamente o tempo de desenvolvimento e solução dos modelos de otimização. Um ambiente de modelagem interativa, linguagem de modelagem completa, e opções flexíveis para tratamento de dados fazem do *LINGO* uma poderosa ferramenta para modelar os problemas mais complexos.

O *LINGO* reconhece variáveis subscritas, conjuntos, operações sobre conjuntos, e expressões matemáticas genéricas. Conseqüentemente os modelos poderão ser expressos de maneira concisa e de fácil leitura. *LINGO* vem com uma biblioteca de funções matemáticas e estatísticas, e permite que os dados sejam armazenados em arquivos externos e planilhas.

Neste capítulo, pretende-se mostrar algumas características do *LINGO* e como o programa foi utilizado no desenvolvimento do presente trabalho.

5.2 COMO ESCREVER UM MODELO NO *LINGO*

Dado um determinado problema, formulado matematicamente, pode-se resolvê-lo, usando o *LINGO*, de três maneiras distintas:

- digitação direta na Janela de Trabalho do *LINGO*;

- através de chamada do executável do *LINGO* por um programa em uma linguagem qualquer;
- através de chamada de uma *DLL* (*Dinamic Linked Library*), cuja forma se pode comparar com a de chamar o *LINGO* na maneira executável; no entanto o *LINGO* é utilizado como um procedimento que está em uma biblioteca do tipo *DLL*. Com este tipo de acesso continua sendo necessária a criação do arquivo com o problema, porém a chamada da rotina é mais rápida e a interface do *LINGO* não aparece na tela, o que pode facilitar para implementações comerciais.

Utilizando um modelo simples para demonstração, apresentar-se-á a sintaxe do *LINGO*.

Supondo o seguinte problema:

Um fabricante de artigos de plástico possui um estoque de 1200 caixas de invólucros transparentes em uma de suas fábricas e outras 1000 caixas em uma segunda fábrica. O fabricante recebeu pedidos deste produto provenientes de três diferentes varejistas nas quantidades de 1000, 700 e 500 caixas, respectivamente. Os custos unitários de expedição desde as fábricas até os varejistas são os seguintes:

TABELA 5.1 – CUSTOS UNITÁRIOS

	Varejista 1	Varejista 2	Varejista 3
Fábrica 1	14	13	11
Fábrica 2	13	13	12

Deseja-se um programa de expedição que atenda todas as demandas a partir do estoque disponível, a um custo mínimo.

Designando-se por $x_{ij}\,(i = 1,2; j = 1,2,3)$ o número de caixas a serem expedidas da fábrica i para o varejista j, o modelo matemático de Programação Linear Inteira para este problema é o seguinte:

$$Min\ z = 14x_{11} + 13x_{12} + 11x_{13} + 13x_{21} + 13x_{22} + 12x_{23}$$

$$s.a. \begin{cases} x_{11} + x_{12} + x_{13} = 1200 \\ + x_{21} + x_{22} + x_{23} = 1000 \\ x_{11} + x_{21} = 1000 \\ x_{12} + x_{23} = 700 \\ x_{13} + x_{23} = 500 \\ x_{ij} \geq 0 \text{ e inteiras} \end{cases}$$

5.2.1 Digitação Direta na Janela de Trabalho do *LINGO*

5.2.1.1 Função Objetivo

Pode-se entrar com cada um dos valores da função objetivo (cada parcela) como é mostrado na figura 5.1 a seguir (o caractere "!" tem a função de inserir um comentário):

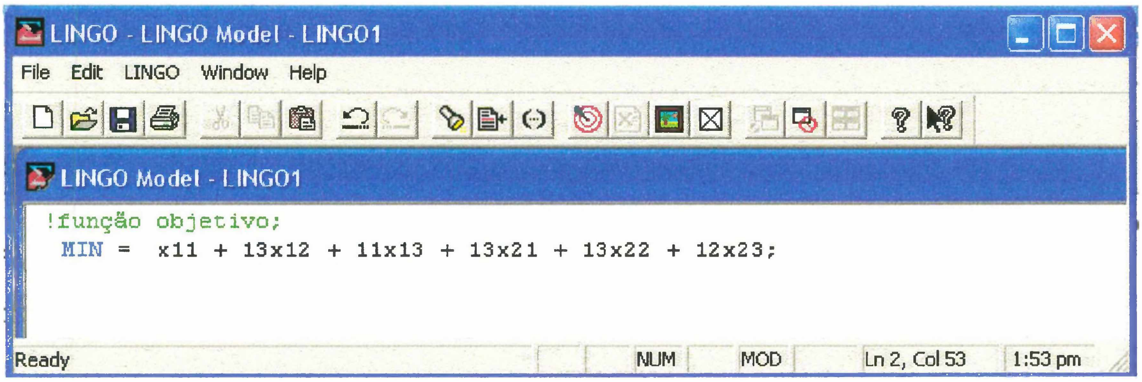


FIGURA 5.1 – FUNÇÃO OBJETIVO NA FORMA DIRETA

No caso de existirem muitos dados, para evitar erros e a digitação de muitos dados, pode-se utilizar o modelo de linguagem do *LINGO*, que é bastante parecida com a linguagem matemática.

Assim, uma expressão da função objetivo pode ser escrita matematicamente como abaixo:

$$Min \sum_{ij} CUSTO_{ij} . QUANTIDADE_{ij} ,$$

onde $CUSTO_{ij}$ pode ser identificado no exemplo como custo de transporte da fábrica *i* para o varejista *j* e QUANTIDADE como o número de caixas a serem expedidas da fábrica *i* ao varejista *j*, e de maneira similar, como pode ser verificado na figura 5.2, na linguagem *LINGO*:

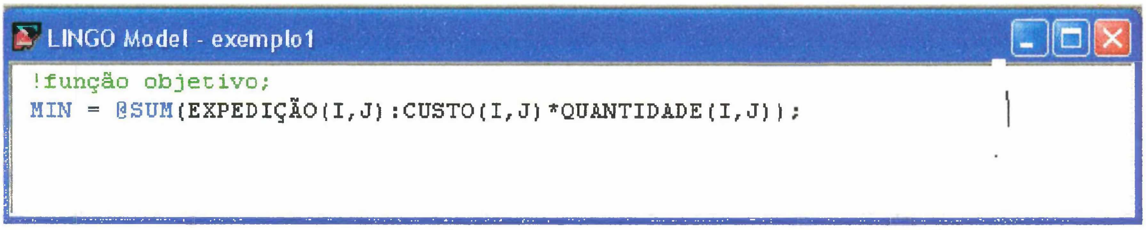


FIGURA 5.2 – FUNÇÃO OBJETIVO COM SINTAXE DO *LINGO*

onde se observa que é necessário criar uma ligação entre custo e quantidade, que no exemplo é representado por EXPEDIÇÃO(I, J). Assim, deseja-se minimizar a soma do custo de expedição multiplicado pelo número de caixas expedidas pelas fábricas aos varejistas em todas as ligações entre fábricas e varejistas. Posteriormente, no item 5.2.1.5, mostra-se a definição de EXPEDIÇÃO no *LINGO*.

Então, a expressão $\sum$ tem seu equivalente no *LINGO* por @SUM.

Existem vários comandos que podem ser utilizados dentro do *LINGO*, entre os mais importantes pode-se citar:

@FOR – para executar laço de índices

@BIN – para definir uma variável como binária

@INT – para definir uma variável como inteira

@FREE – para definir uma variável irrestrita de sinal

5.2.1.2 As Restrições

No exemplo que se está analisando, existem cinco restrições, onde as duas primeiras garantem que as quantidades expedidas não ultrapassem a disponibilidade em estoque em cada fábrica, e as três últimas garantem que não se ultrapasse aos pedidos de cada varejista.

Para passar para a linguagem *LINGO*, deve-se nomear cada conjunto de restrições. Assim, como se observa na figura 5.3, pode-se escrever:

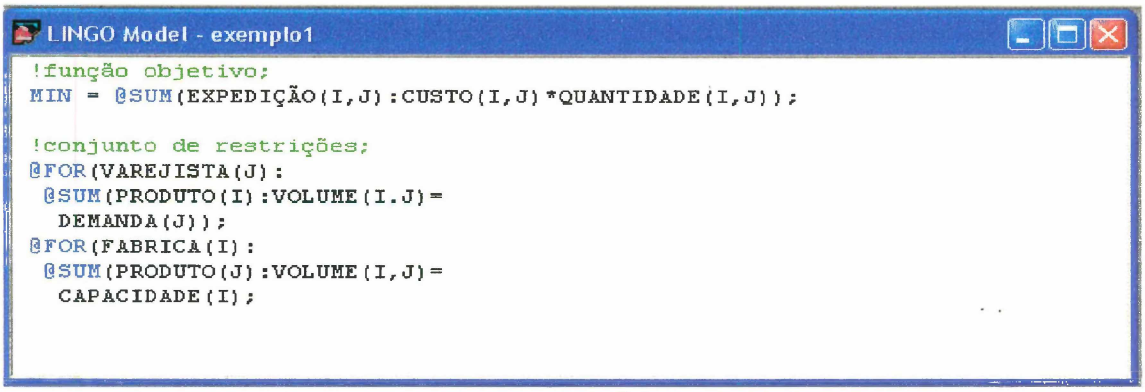


FIGURA 5.3 – SINTAXE DO *LINGO* PARA AS RESTRIÇÕES

Pode-se observar, neste pequeno exemplo, algumas exigências sintáticas:

- o ponto-e-vírgula no final de cada linha é obrigatório. Sem sua utilização o *LINGO* não procederá à resolução do problema;
- "MIN =" é a definição para que a função objetivo seja minimizada. Para a maximização deve-se utilizar "MAX =";
- utilização do asterisco para efetuar a multiplicação;
- as variáveis podem ser escritas com letras maiúsculas ou minúsculas, pois o *LINGO* considerará X = x.

Vale a pena salientar que as variáveis podem conter até 32 caracteres, que podem ser letras (A-Z), números (1-9) ou o sublinhado (_). Qualquer outro símbolo não pode ser utilizado.

5.2.1.3 Definição dos Conjuntos de Trabalho

A utilização de variáveis indexadas (vetores e matrizes) é imprescindível em grandes problemas. Estas variáveis devem ser definidas, assim como os valores dos índices que elas poderão assumir. A sintaxe utilizada no *LINGO* é a seguinte:

```
SETS:
        definições
ENDSETS
```

Supondo que um problema utilize uma variável indexada X variando de 1 a 100, tem-se: X(1), X(2), X(3),...,X(100). Chamando este índice que varia de 1 a 100 de VAR, a sintaxe para definir esta variável fica:

VAR / 1 .. 100 /: X;

Se no mesmo problema existirem coeficientes C variando de 1 a 100, pode-se utilizar a mesma linha de comando para defini-los (observar que as constantes podem ser definidas junto com as variáveis), resultando em:

VAR / 1 .. 100 /: X,C;

Para uma outra constante, por exemplo N variando de 1 até 50, é necessária outra linha de definição, assim:

COL / 1 .. 50 /: N:

Agora COL é o índice associado à variação de 1 a 50. A sintaxe completa fica da seguinte maneira:

```
SETS:
    VAR /1 .. 100/ : X,C;
    COL / 1 .. 50 / : N;
ENDSETS
```

5.2.1.4 Definição de Dados

Depois de definidos os conjuntos de trabalho, é necessário entrar com os valores dos coeficientes constantes, para isto utiliza-se outra parte para dados, cuja sintaxe é:

```
DATA:
    dados
ENDDATA
```

Supondo que seja necessário entrar com os 100 valores de C, para isto bastaria escrever:

C = 1, 2, 4, 12, 32, 45...;

(onde se escrevem os 100 valores correspondentes, separados por vírgulas. Fica claro que as reticências não fazem parte da sintaxe).

Se todos os valores fossem iguais, por exemplo, os 50 valores de N fossem iguais a 3, pode-se escrever:

N = 3, 3, 3, 3,...;

(repetindo-se o valor 3 cinqüenta vezes), ou, simplesmente:

N = 3;

Assim a sintaxe completa fica:

```
DATA:
    C = 1, 2, 4, 12, 32, 45,...;
    N = 3;
ENDDATA
```

5.2.1.5 Modelo Completo

Aplicando as definições no exemplo deste capítulo, observa-se pela figura 5.4, como o problema fica escrito de forma completa:

LINGO Command Script - exemplo1

```
MODEL:
!conjuntos de trabalho;
SETS:
      FABRICAS /F1 F2/: CAPACIDADE;
      VAREJISTAS/V1 V2 V3/: DEMANDA;
      EXPEDICAO(FABRICAS, VAREJISTAS): CUSTO, QUANTIDADE;
ENDSETS
!dados do problema;
DATA:
      CAPACIDADE = 1200 1000;
      DEMANDA = 1000 700 500;
      CUSTO = 14, 13, 11,
              13, 13, 12;
ENDDATA

!função objetivo;
MIN = @SUM(EXPEDICAO(I,J):CUSTO(I,J)*QUANTIDADE(I,J));

!conjunto de restrições;
@FOR(VAREJISTAS(J):
 @SUM(FABRICAS(I):QUANTIDADE(I,J))=
  DEMANDA(J));
@FOR(FABRICAS(I):
 @SUM(VAREJISTAS(J):QUANTIDADE(I,J))=
  CAPACIDADE(I));
END
```

FIGURA 5.4 – MODELO COMPLETO

Assim, utilizando-se a sintaxe do *LINGO*, o exemplo pode ser dividido em várias partes, cada uma com sua utilização específica. Para se definir um modelo, deve-se basicamente seguir o seguinte esquema:

MODEL:
    (definição dos conjuntos)
    (determinação dos dados)
    (formulação do problema)
END

O comando MODEL indica ao *LINGO* que se está pronto para entrar com o modelo, e o comando END que o modelo introduzido já acabou.

Observar que se utilizam dois pontos no final de SETS e DATA e que não se utiliza sinal após END, ENDSETS e ENDDATA.

Após terminado o programa, clicando-se no ícone , ou em EDIT seguido de SOLVE, aparece a janela apresentada na figura 5.5 a seguir:

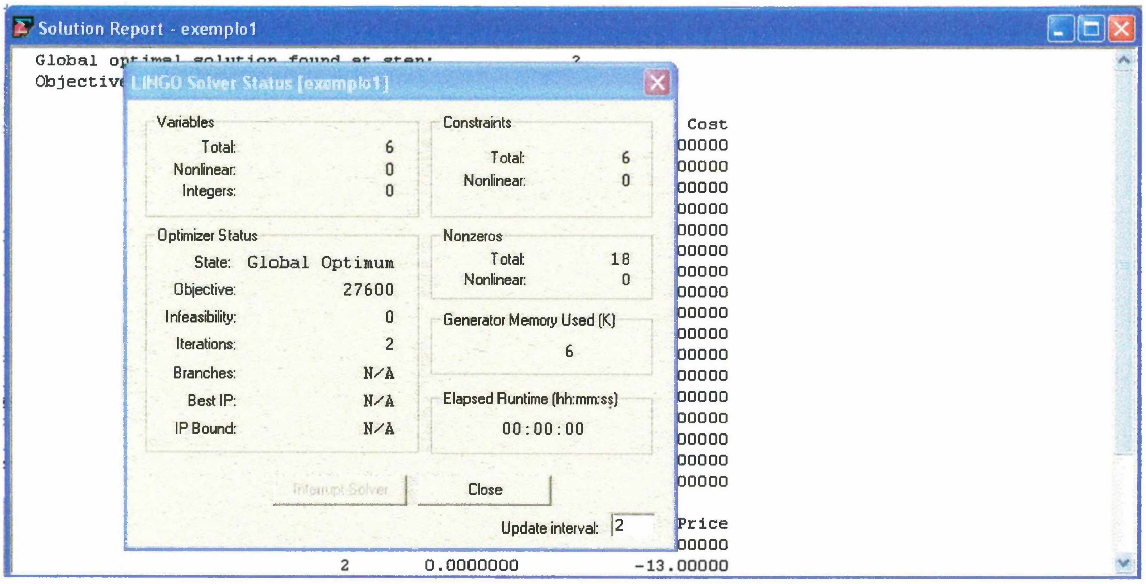


FIGURA 5.5 – JANELA: SOLUÇÃO

e clicando sobre CLOSE, temos a solução do problema, como mostra a figura 5.6:

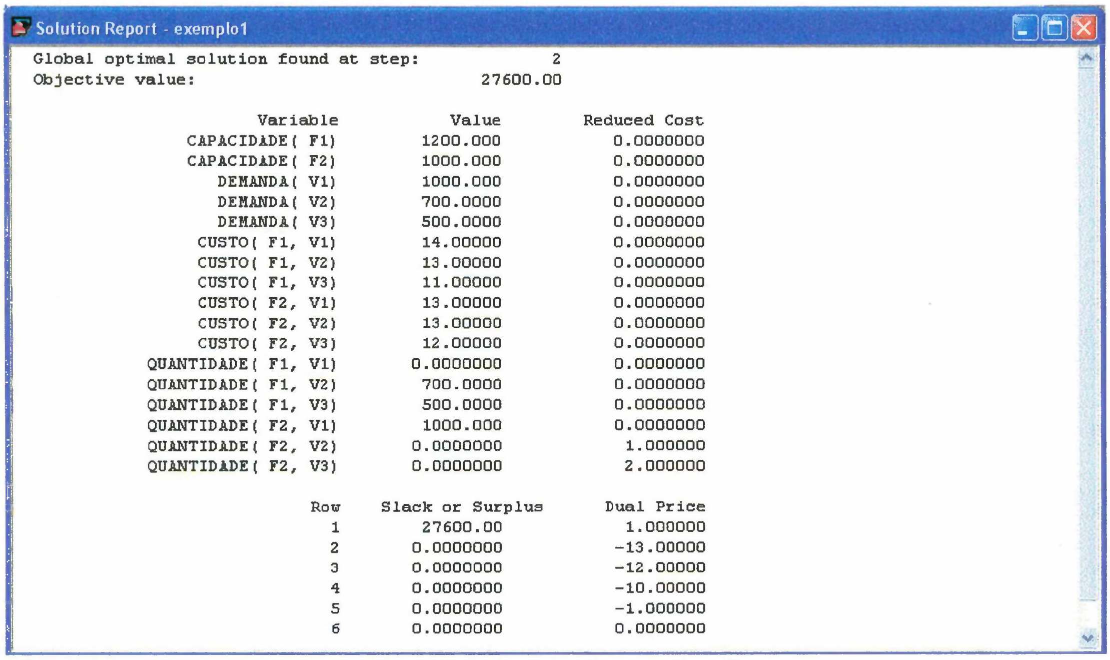


FIGURA 5.6 – JANELA: SOLUÇÃO GLOBAL

Assim sendo, a solução ótima para o problema será transportar 700 caixas da fábrica 1 ao varejista 2; 500 caixas da fábrica 1 ao varejista 3 e 1000 caixas da fábrica 2 ao varejista 1, com custo total do transporte de $ 27 600.

5.2.2 Arquivo de Chamada do *LINGO*

Observar a seguinte seqüência de comandos:

SET TERSEO 1
!Comando para desabilitar a saída no vídeo após a resolução;

GO
!Comando para mandar executar o *LINGO*;

DIVERT \caminho\arquivo
!Comando que redireciona a saída para determinado arquivo;

NONZ X
!Comando que manda salvar no arquivo de saída as variáveis X não zeradas;

RVRT
!Comando que fecha o arquivo de saída;

QUIT
!Comando para sair do *LINGO*;

Estes comandos, que definirão os parâmetros de resolução, deverão ser salvos em um arquivo com uma extensão "LTF" que não poderá ser executado diretamente na tela de digitação do *LINGO*, devendo-se chamá-lo através de um outro programa.

Esta chamada é executada de maneiras diferentes, dependendo da linguagem utilizada. O comando direto a ser digitado no DOS para chamar o *LINGO* é:

***LINGO6**LINGO*.EXE –Tnomedoarquivo**

onde "*nomedoarquivo*" é o arquivo salvo com as definições do problema de modo completo, ou seja: deve-se indicar o drive e diretório onde o mesmo se encontra.

Para utilizar esta chamada, o arquivo deverá ser digitado após o END, assim, o esquema fica da seguinte maneira:

MODEL:
(definição dos conjuntos)
(determinação dos dados)
(formulação do problema)
END
Definição de parâmetros de resolução

5.2.3 Estrutura de Funcionamento para a Chamada do *LINGO* por um Programa Executável

A seguir, é apresentado na figura 5.7 o fluxograma que representa a implementação computacional da resolução através do *LINGO* :

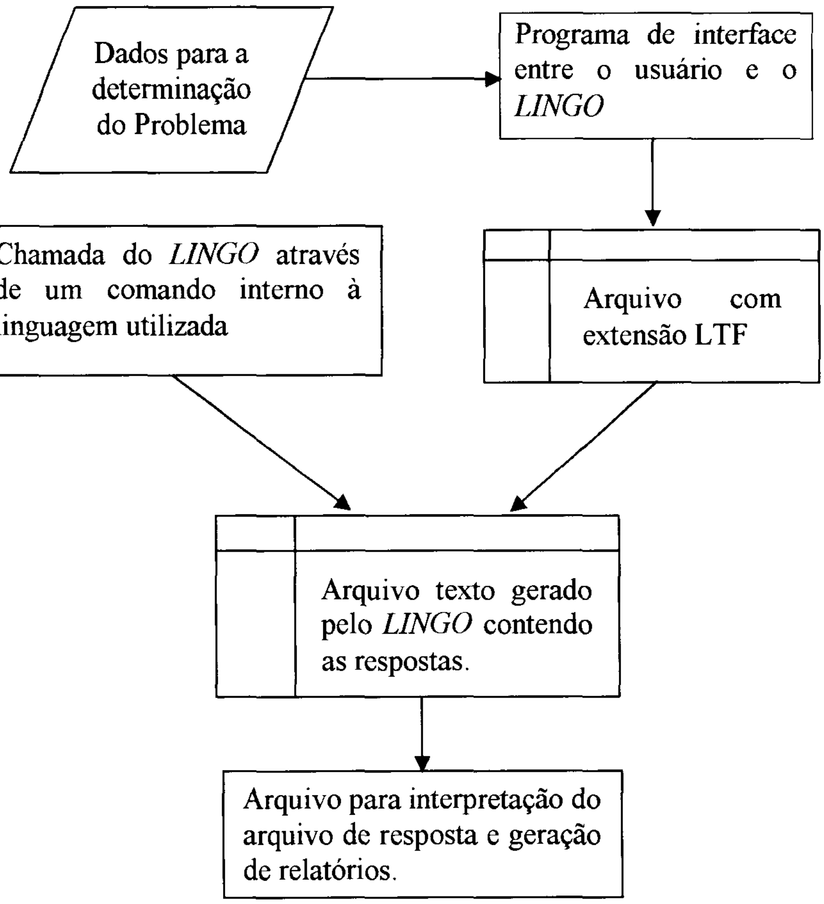


FIGURA 5.7 – FLUXOGRAMA PARA CHAMADA *LINGO* POR PROGRAMA EXECUTÁVEL

No próximo capítulo será apresentado o problema de escala de plantão para serviço de guarda, abordado neste trabalho, com a formulação matemática, e após, a implementação computacional, onde foi utilizado o *software LINGO*.

CAPÍTULO VI

6 O PROBLEMA DE ESCALA PARA SERVIÇO DE GUARDA

6.1 INTRODUÇÃO

Para compor a escala de serviço de guarda foram considerados o número de militares necessário por dia e as quantidades disponíveis, se existia ou não carga especial de trabalho para determinados militares e o tipo de escala necessária para o mês. Depois, foi verificado o problema dos finais de semana, para evitar o trabalho em finais de semana seguidos e também para evitar que fosse ultrapassado o número máximo de finais de semana a serem trabalhados no mês.

Para evitar que um militar viesse a trabalhar dias seguidos no plantão ou que alguma das restrições fossem violadas quando se passasse de um mês para o outro, foi criado um vínculo com o mês atual, em relação ao último dia trabalhado e o último final de semana trabalhado pelo militar no mês anterior.

Considerando a hierarquia dos militares, foi determinada a função objetivo, que é a de maximizar o nível de escolha dos dias em que os militares gostariam de trabalhar no plantão.

6.2 FORMULAÇÃO MATEMÁTICA PARA O PROBLEMA

A seguir será feita a formulação matemática genérica do problema, sempre precedida de um exemplo numérico para melhor compreensão.

6.2.1 Cálculos Preliminares

Supõe-se a necessidade de 15 militares/dia no plantão num determinado mês de 30 dias. Esta quantidade pode sofrer alteração em um determinado dia, devido a problemas de demanda, como uma visita oficial de uma autoridade, por exemplo.

Denota-se a necessidade diária de militares por $\mathbf{d_j}$, e o número de dias do período a ser analisado por **nd**.

Assim a necessidade de militares em trabalho durante este período de 30 dias é de $15 \cdot 30 = 450$, ou genericamente:

$$\begin{aligned} &militares/dia = \sum_{j=1}^{nd} d_j \\ &d_j = \text{demanda do dia } j \\ &nd = \text{número de dias.} \end{aligned} \tag{6.1}$$

No caso de não haver sido estabelecida uma carga especial de trabalho no plantão para os militares, e considerando disponíveis 70 militares para cobrir esta demanda, tem-se:

$$\frac{450}{70} = 6{,}04$$

Assim sendo, a quantidade mínima de dias trabalhados no plantão para cada militar é de 6 dias. Genericamente tem-se a determinação do mínimo a ser trabalhado por:

$$mdt = \left\lfloor \frac{\sum_{j=1}^{nd} d_j}{nm} \right\rfloor \tag{6.2}$$

nm = número de militares

mdt = mínimo de dias trabalhados no plantão, onde $\lfloor b \rfloor$ é o maior inteiro menor ou igual a b

Chamando de $\mathbf{q_i}$ a quantidade de dias trabalhados pelo militar **i**, tem-se:

$$\sum_{i=1}^{nm} q_i = \sum_{j=1}^{nd} d_j\,, \tag{6.3}$$

mas

$$nm \cdot mdt \leq \sum_{j=1}^{nd} d_j \,. \tag{6.4}$$

Assim sendo, a quantidade de militares que trabalharão um dia a mais **(qmt)** é dada pela expressão:

$$qmt = \sum_{j=1}^{nd} d_j - (nm \cdot mdt) \tag{6.5}$$

Neste exemplo tem-se, portanto, 40 militares trabalhando 6 dias/mês e 30 militares trabalhando 7 dias/mês, ou seja, $40 \cdot 6 + 30 \cdot 7 = 450$, que é igual a demanda total.

Com as informações anteriores pode-se estabelecer um critério de pré-viabilidade do problema, em função da escala escolhida. Ou seja, se a escala for de 24/96, tem-se que a cada 5 dias apenas um é trabalhado pelo mesmo militar. Assim sendo em 30 dias trabalhados cada militar pode trabalhar um máximo de 6 dias. Com esta escala o exemplo dado fica inviável, pois como foi visto, alguns militares devem trabalhar 7 dias.

No entanto, para uma escala de 24/72, tem-se que a cada quatro dias um é trabalhado pelo mesmo militar, ou 25% . De 30 dias tem-se um limite máximo de 7,5 dias, ou seja, na prática um limite de 7 dias no mês, o que implicará em uma relativa *folga* que poderá viabilizar as demais restrições, mas sem a garantia da viabilidade.

No caso de existirem cargas especiais de trabalho para determinados militares, reduzir-se-á da carga total necessária a somatória das cargas especiais. Sejam $E = \{i_1, i_2, ..i_m\}$ o conjunto dos militares que possuem carga especial de trabalho, onde *m* é a cardinalidade de *E*, e $CG = \{q_{i1}, q_{i2}, ..., q_{im}\}$ o conjunto de suas cargas correspondentes, com $m \leq nm$. Assim, tem-se a expressão:

$$\text{nova carga} = \sum_{j=1}^{nm} d_j - \sum_{i \in E} q_i \tag{6.6}$$

Nesta nova carga serão considerados *nm – m* militares, e, conseqüentemente, para uma análise da carga dos militares restantes, ter-se-á:

$$militares\_dia = \sum_{j=1}^{nm} d_j - \sum_{i \in E} q_i \tag{6.7}$$

$$mdt = \left\lfloor \frac{\sum_{j=1}^{nm} d_j - \sum_{i \in E} q_i}{nm - m} \right\rfloor \tag{6.8}$$

$$(nm - m)\, mdt \leq \sum_{j=1}^{nm} d_j - \sum q_i \tag{6.9}$$

$$qmt = \left(\sum_{j=1}^{nm} d_j - \sum_{i \in E} q_i \right) - (nm - m)\, mdt\,. \tag{6.10}$$

6.2.2 Variáveis de Decisão

As variáveis de decisão para o problema em questão podem ser definidas através das variáveis binárias:

$$x_{ij} = \begin{cases} 1 & \text{se o militar } i \text{ trabalha no plantão no dia } j; \\ 0 & \text{caso contrário.} \end{cases}$$

6.2.3 Restrições para o Problema

As restrições de oferta podem ser escritas como:

$$\sum_{j=1}^{nd} x_{ij} = q_i \qquad i = 1, \ldots, nm \tag{6.11}$$

e as restrições de demanda por:

$$\sum_{i=1}^{nm} x_{ij} = d_j \qquad j = 1, \ldots, nd\,. \tag{6.12}$$

Não existe mudança alguma na formulação se as equações das restrições (6.11) e (6.12) forem substituídas por inequações. Sabe-se que a igualdade (6.3) verifica-se, então se pode substituir (6.11) por

$$\sum_{j=1}^{nd} x_{ij} \leq q_i \qquad i = 1,\ldots,nm \tag{6.13}$$

e com isto, garante-se que nenhum militar irá trabalhar mais que sua carga limite de plantão e substitui-se a equação (6.12) por

$$\sum_{i=1}^{nm} x_{ij} \geq d_j \qquad j = 1,\ldots,nd\,, \qquad 6.14)$$

garantindo-se que a demanda será atendida em todos os dias.

Para as restrições de escala, tem-se que em uma escala 24/72, em um mês de 30 dias, para que não sejam considerados os dias do mês seguinte, deve-se fazer a operação $30-4+1=27$. Ou seja, o último dia auxiliar de início de comparação a se considerar será o dia 27. Para uma escala 24/96, ou em cada 5 dias um é trabalhado, tem-se $30-5+1=26$ e para uma escala 24/48, onde a cada 3 dias um é trabalhado, tem-se $30-3+1=28$.

Considerando A/B os tipos de escala, com A = 24h e B as horas de folga (24, 48, 72,...), e

$$nd_1 = \frac{A+B}{24} \tag{6.15}$$

o fator de escala, e supondo um militar **i** que começa no primeiro dia do mês na escala 24/72, então temos as restrições:

$$\begin{gathered} x_{i1} + x_{i2} + x_{i3} + x_{i4} \leq 1 \\ x_{i2} + x_{i3} + x_{i4} + x_{i5} \leq 1 \\ \vdots \\ x_{i27} + x_{i28} + x_{i29} + x_{i30} \leq 1, \end{gathered} \tag{6.16}$$

as quais impedem que um militar trabalhe mais de 1 dia em 4 dias consecutivos.

Generalizando, para um militar i, começando em um dia j, tem-se:

$$\sum_{k=1}^{nd_1} x_{i,j+k-1} \leq 1 \qquad \begin{array}{l} i = 1,\ldots,nm \\ j = 1,\ldots,nd - nd_1 + 1. \end{array} \tag{6.17}$$

Agora o problema é o do equacionamento dos finais de semanas. Para isto, será utilizado um exemplo considerando como primeiro sábado do mês o dia 5, que será generalizado posteriormente.

Se o primeiro sábado do mês for dia 5, tem-se a tabela 6.1 a seguir:

TABELA 6.1 – FINAIS DE SEMANA

Sábado	Domingo
5	6
12	13
19	20
26	27

ou de uma maneira genérica, sendo o primeiro sábado do mês dado por S, temos a tabela 6.2 a seguir:

TABELA 6.2 – GENERALIZAÇÃO PARA FINAIS DE SEMANA

Sábado	Domingo
-	S – 6
S	S + 1
S + 7	S + 8
S + 14	S + 15
S + 21	S + 22
S + 28	S + 29

desde que $(1 \leq S + n \leq nd)$.

Para escrever, por exemplo, que o militar **i** só pode trabalhar um domingo no mês, usa-se:

$$x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} \leq 1 \tag{6.18}$$

ou, no exemplo considerado:

$$x_{i,6} + x_{i,13} + x_{i,20} + x_{i,27} \leq 1 .$$

Sendo **nsab** o número máximo de sábados que o militar **i** pode trabalhar no mês, então:

$$x_{i,S} + x_{i,S+7} + x_{i,S+8} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} \le nsab\,. \tag{6.19}$$

Sendo **ndom** o número máximo de domingos que o militar **i** pode trabalhar no mês, então:

$$x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} \le ndom\,. \tag{6.20}$$

Sendo **nfim** o número máximo de finais de semana que o militar **i** pode trabalhar no mês, então:

$$\begin{aligned} & x_{i,S} + x_{i,S+7} + x_{i,S+8} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + \\ & + x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} \le nfim \end{aligned} \tag{6.21}$$

Também se faz necessária uma restrição que inviabilize o trabalho em finais de semana consecutivos. Isto pode ser obtido através das seguintes restrições:

$$\begin{aligned} & x_{i,S-6} + x_{i,S} + x_{i,S+1} \le 1 \\ & x_{i,S} + x_{i,S+1} + x_{i,S+7} + x_{i,S+8} \le 1 \\ & \quad \vdots \\ & x_{i,S+21} + x_{i,S+22} + x_{i,S+28} + x_{i,S+29} \le 1. \end{aligned} \qquad 6.22)$$

Entretanto, não se pode analisar cada mês de uma maneira estática, pois cada militar já vem de um determinado esquema anterior de trabalho no plantão que deve ser considerado na formulação.

Então existe a necessidade de um vínculo com o mês anterior que será obtido através das informações:

- último dia trabalhado no mês anterior;
- último final de semana do mês anterior.

Pode-se observar através da tabela 6.3 abaixo, que considera uma escala de 24/72, ou um dia trabalhado em 4, o primeiro dia do mês que o militar poderia cumprir guarda, considerando o último dia em que cumpriu guarda no mês anterior:

TABELA 6.3 – VÍNCULO ENTRE MÊS ANTERIOR E MÊS ATUAL

Mês anterior				Mês atual			
27	28	29	30	1	2	3	4
			X				O
		X				O	
	X				O		
X				O			

X indica o último dia trabalhado no mês anterior
O indica o dia a partir do qual poderá trabalhar

Assim, se o militar cumpriu guarda no dia 27 do mês anterior, poderá cumprir guarda a partir do primeiro dia do mês atual; se seu último dia de guarda no mês anterior foi o dia 28, só poderá cumprir guarda a partir do dia 2 do mês atual e assim por diante.

Assim sendo, se o militar trabalhou no último dia do mês anterior, o primeiro dia que ele poderá trabalhar será dia 4, ou seja, não poderá trabalhar nos 3 primeiros dias do mês, ou seja:

$$\sum_{j=1}^{3} x_{ij} = 0 \quad \text{para o militar } i. \tag{6.23}$$

O mesmo raciocínio pode ser repetido para outros dias. Se o militar trabalhou há mais de quatro dias, porém, nenhuma restrição adicional deve ser imposta devido ao tipo de escala considerado.

Assim, seja **udt** o valor que representa a diferença entre o último dia do mês e o último dia trabalhado pelo militar.

Se o militar trabalhou no último dia do mês anterior, udt = 0; se trabalhou no penúltimo dia, udt = 1, e assim sucessivamente.

Considerando o fator de escala (6.15), se $udt \geq nd_1 - 1$, então nenhuma restrição adicional deve ser considerada. Caso contrário, se $udt < nd_1 - 1$, então a restrição:

$$\sum_{j=1}^{nd_1 - 1 - udt} x_{ij} = 0 \text{ para } i = 1, \ldots, nm, \quad \text{se} \quad udt < nd_1 - 1 \tag{6.24}$$

deve ser considerada.

Assim, se:

$nd_1 = 4$

Se $udt = 0 \Rightarrow \sum_{j=1}^{3} x_{ij} = 0$

Se $udt = 1 \Rightarrow \sum_{j=1}^{2} x_{ij} = 0$

Se $udt = 2 \Rightarrow \sum_{j=1}^{1} x_{ij} = 0$

Se $udt = 3 \Rightarrow$ não existe restrição

Um raciocínio semelhante deve ser utilizado para considerar os finais de semana, porém só será levado em consideração o fato do militar ter ou não trabalhado no último final de semana do mês anterior, visto que a quantidade de finais de semana trabalhados está restrita unicamente ao mês.

Faz-se simplesmente a pergunta: o militar **i** trabalhou o último final de semana do mês anterior? Se a resposta for afirmativa, deve-se incluir a restrição (6.25) abaixo:

$$x_{i,S} + x_{i,S+1} = 0 \quad \text{para o militar } i, \tag{6.25}$$

mesmo quando o primeiro sábado do mês cair no dia 7. A garantia que não haverá trabalho para este militar no dia 1° (domingo) é dada pela restrição (6.24).

6.2.4 Função Objetivo para o Problema

Até o momento não se considerou a função objetivo.

Com base no que já foi citado em relação à hierarquia militar, a preferência deve ser dada aos militares mais antigos.

A cada militar estará associado um peso (**p_i**) que representará o grau de prioridade em relação aos demais, ou seja, tanto maior será este peso, quanto mais antigo for o militar. Tal peso, escolhido empiricamente com variação linear, será utilizado na função objetivo multiplicando por uma determinada preferência diária (ver definição de n_{ij} abaixo).

Cada militar indica **qEs** dias nos quais gostaria de trabalhar no plantão e uma outra quantidade **qEx** de dias que não gostaria de trabalhar no plantão.

Como o militar trabalha poucos dias no plantão durante o mês, normalmente, a quantidade de dias de escolha para trabalhar será menor que a quantidade de escolha de dias para não se trabalhar no plantão.

Considerando os valores para a preferência

$$n_{ij} = \begin{cases} 1, & \text{se o militar } i \text{ escolhe o dia } j \text{ para trabalhar no plantão;} \\ 0, & \text{se é indiferente;} \\ -1, & \text{se o dia } j \text{ é inconveniente para o militar } i, \end{cases}$$

a expressão

$$x_{ij} \cdot p_i \cdot n_{ij}, \tag{6.26}$$

representará um nível de escolha do militar ***i*** em relação ao dia ***j*** considerando sua hierarquia dentro da estrutura.

O objetivo é maximizar este nível de escolha. Assim a função objetivo é:

$$\max \quad \sum_{i=1}^{nm} \sum_{j=1}^{nd} x_{ij} \cdot p_i \cdot n_{ij} \tag{6.27}$$

6.2.5 Formulação do Problema Completo

Assim sendo, a formulação completa para o problema fica definido da seguinte forma,

sendo $x_{ij} = \begin{cases} 1, & \text{se o militar } i \text{ trabalha no plantão no dia } j; \\ 0, & \text{caso contrário.} \end{cases}$

$$\max \quad \sum_{i=1}^{nm} \sum_{j=1}^{nd} x_{ij} \cdot p_i \cdot n_{ij}$$

s.a

$$\sum_{j=1}^{nd} x_{ij} \le q_i \qquad i = 1...nm$$

$$\sum_{i=1}^{nm} x_{ij} \ge d_j \qquad j = 1...nd$$

$$\sum_{k=1}^{nd_1} x_{i,j+k-1} \le 1 \qquad \begin{matrix} i = 1,\dots,nm \\ j = 1,\dots,nd - nd_1 + 1 \end{matrix}$$

$$x_{i,S} + x_{i,S+7} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} \le nsab$$

$$x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} \le ndom$$

$$\begin{cases} x_{i,S} + x_{i,S+7} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + \\ +x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} \le nfim \end{cases}$$

$$x_{i,S-6} + x_{i,S} + x_{i,S+1} \le 1$$

$$x_{i,S} + x_{i,S+1} + x_{i,S+7} + x_{i,S+8} \le 1$$

$$\vdots$$

$$x_{i,S+21} + x_{i,S+22} + x_{i,S+28} + x_{i,S+29} \le 1$$

$$\sum_{j=1}^{nd_1-1-udt} x_{ij} = 0 \text{ para } i = 1,\dots,nm \text{ se } udt < nd_1 - 1$$

$$x_{i,S} + x_{i,S+1} = 0 \quad \text{se o militar i trabalhou no último fim de semana do mês anterior}$$

Tem-se $nd \cdot nm$ variáveis, e um número de restrições no mínimo igual a: $nm + nd + (nd - nd_1 + 1)nm + 3nm \cdot (\text{número de finais de semana})$.

6.3 ALTERAÇÃO PÓS-OTIMIZAÇÃO

Um problema real a ser considerado é o fato de um determinado militar faltar após ter sido feita a escala dos militares. Quando isto ocorre, visto que a exigência da quantidade de militares dever ser cumprida, a solução atualmente praticada é a de fazer um dos militares que deveria sair de serviço permanecer em plantão por mais um dia. Tal situação é extremamente desgastante para o militar que é retido para cobrir este serviço. A única compensação que o militar recebe é a de cumprir guarda um dia a menos no mês seguinte.

Uma outra alteração possível de ocorrer é a falta de um militar por um longo período de tempo, provocada por afastamento para treinamento ou por motivo de doença. Quando isto ocorre toda a escala deve ser refeita, e, conseqüentemente muitos militares terão seus horários modificados por esta alteração não prevista.

Para que a tabela de plantões seja refeita, uma possibilidade é a de resolver um problema de determinação de escala para um número menor de militares, excluindo-se os ausentes por necessidade. O programa seria então executado somente para o período restante, com os pesos atribuídos a cada militar iguais, e como escolha os dias em que o militar foi indicado para trabalhar e como não escolha os dias marcados para não trabalhar na determinação do primeiro problema. Isto resolve o problema de minimizar o número de trocas.

Claramente estas alterações, por menores que sejam, implicariam em alterações na tabela já existente implicando, conseqüentemente, em uma nova distribuição pessoal de encargos para cada funcionário.

6.3.1 Escala com Folga para Suprimento de Faltas

Com o objetivo de facilitar o gerenciamento de faltas de pessoal, foi sugerida aos militares encarregados das escalas uma estratégia diferenciada para a execução das mesmas, descrita a seguir.

A cada dia a demanda seria aumentada em um valor $\mathbf{e_j}$ (inteiro), desde que este valor possa ser viável para se cobrir o serviço de plantão com os militares disponíveis. O militar que fica a disposição para cobrir esta demanda é dito de **sobreaviso**.

No exemplo considerado, com necessidade de 15 militares dias, coloca-se uma demanda extra de 2 militares por dia. A demanda total diária seria então de 17 militares.

O problema seria resolvido para esta nova demanda diária ($\mathbf{d_j} + \mathbf{e_j}$, no exemplo 15 + 2), existindo então a possibilidade de uma cobertura extra de militares, caso alguém faltasse.

No caso de uma falta individual por um único dia, o militar mais novo em sobreaviso seria chamado, assim como no caso de falta prolongada. O segundo militar de sobreaviso no dia seria chamado somente se houvesse outra falta no mesmo dia.

O militar que está de sobreaviso naquele dia não ganha a vantagem de não trabalhar o horário administrativo do dia seguinte, sendo que só terá direito à dispensa se efetivamente trabalhar em plantão, ao cobrir um faltante.

6.3.2 Variáveis de Decisão para o Sobreaviso

As variáveis de decisão para esta variação do problema podem ser definidas através das seguintes variáveis binárias:

$$y_{ij} = \begin{cases} 1, & \text{se o militar } i \text{ está de sobreaviso no dia } j; \\ 0, & \text{caso contrário.} \end{cases}$$

6.3.3 Restrições Para o Problema com Sobreaviso

As restrições de oferta podem ser escritas, com as características do problema de sobreaviso, pela expressão (6.13) e:

$$\sum_{j=1}^{nd} y_{ij} \le r_i \qquad i = 1,\ldots,nm, \qquad 6.28)$$

onde r_i é a oferta extra incluída para cada militar para poder ser possível atender às novas demandas de sobreaviso; e as restrições de demanda pela equação (6.14) e por:

$$\sum_{i=1}^{nm} y_{ij} \ge e_j \qquad j = 1,\ldots,nd. \qquad (6.29)$$

Nas restrições que garantem que a escala será obedecida (6.17) deve-se incluir as variáveis de sobreaviso, ou seja:

$$\sum_{k=1}^{nd_1} x_{i,j+k-1} + \sum_{k=1}^{nd_1} y_{i,j+k-1} \le 1 \qquad \begin{array}{l} i = 1,\ldots,nm \\ j = 1,\ldots,nd - nd_1 + 1. \end{array} \qquad (6.30)$$

De forma análoga para as expressões (6.18) e (6.19), o militar **i** só pode trabalhar **ndom** domingos e **nsab** sábados no mês, então:

$$\begin{array}{l} x_{i,S} + x_{i,S+7} + x_{i,S+8} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + \\ + y_{i,S} + y_{i,S+7} + y_{i,S+8} + y_{i,S+14} + y_{i,S+21} + y_{i,S+28} \le nsab \end{array} \qquad (6.31)$$

$$\begin{array}{l} x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} + \\ + y_{i,S-6} + y_{i,S+1} + y_{i,S+8} + y_{i,S+15} + y_{i,S+22} + y_{i,S+29} \le ndom. \end{array} \qquad (6.32)$$

E no lugar de (6.20) sendo **nfim** o número máximo de finais de semana que o militar **i** pode trabalhar no mês, então:

$$\begin{array}{l} x_{i,S} + x_{i,S+7} + x_{i,S+8} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + \\ + x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} + \\ y_{i,S} + y_{i,S+7} + y_{i,S+8} + y_{i,S+14} + y_{i,S+21} + y_{i,S+28} + \\ + y_{i,S-6} + y_{i,S+1} + y_{i,S+8} + y_{i,S+15} + y_{i,S+22} + y_{i,S+29} \le nfim. \end{array} \qquad (6.33)$$

Quanto à restrição que inviabilize o trabalho em finais de semana consecutivos, tem-se substituindo (6.21):

$$
\begin{aligned}
& x_{i,S-6} + x_{i,S} + x_{i,S+1} + y_{i,S-6} + y_{i,S} + y_{i,S+1} \le 1 \\
& x_{i,S} + x_{i,S+1} + x_{i,S+7} + x_{i,S+8} + y_{i,S} + y_{i,S+1} + y_{i,S+7} + y_{i,S+8} \le 1 \\
& \quad \vdots \\
& x_{i,S+21} + x_{i,S+22} + x_{i,S+28} + x_{i,S+29} + y_{i,S+21} + y_{i,S+22} + y_{i,S+28} + y_{i,S+29} \le 1.
\end{aligned}
\qquad (6.34)
$$

Para as restrições que vinculam o mês atual ao mês anterior substitui-se (6.22) por:

$$\sum_{j=1}^{nd_1-1-udt} x_{ij} + \sum_{j=1}^{nd_1-1-udt} y_{ij} = 0, \text{ para } i = 1,\ldots,nm, \quad \text{se} \quad udt < nd_1 - 1 \qquad (6.35)$$

e se o militar **i** trabalhou no último final de semana do mês anterior, substitui-se a equação (6.25) por:

$$x_{i,S} + x_{i,S+1} + y_{i,S} + y_{i,S+1} = 0 \quad \text{para o militar } i \qquad (6.36)$$

6.3.4 Função Objetivo para o Problema com Sobreaviso

Consideram-se os mesmos pesos individuais associados a cada militar e o mesmo critério de escolha como no item 6.2.4 adicionando-se a (6.26) a parcela:

$$y_{ij} \cdot p_i \cdot n_{ij}^*, \qquad (6.36)$$

que representa um nível de escolha para sobreaviso do militar **i** em relação ao dia **j** considerando sua hierarquia dentro da estrutura. O objetivo é maximizar este nível de escolha. Assim a função objetivo fica:

$$\max \quad \sum_{i=1}^{nm} \sum_{j=1}^{nd} x_{ij} \cdot p_i \cdot n_{ij} + \sum_{i=1}^{nm} \sum_{j=1}^{nd} y_{ij} \cdot p_i \cdot n_{ij}^* \qquad (6.37)$$

Poder-se-ia estabelecer novos critérios de escolha para esta metodologia de sobreaviso, porém no exemplo os valores tomados foram os mesmos, $n_{ij}^* = n_{ij}$.

6.3.5 Formulação Completa do Problema com Sobreaviso

Assim sendo, a formulação completa considerando-se o problema com sobreaviso, sendo

$$x_{ij} = \begin{cases} 1 & \text{se o militar } i \text{ trabalha no plantão no dia } j \\ 0 & \text{caso contrário} \end{cases}$$ e

$$y_{ij} = \begin{cases} 1 & \text{se o militar } i \text{ está de sobreaviso no dia } j \\ 0 & \text{caso contrário} \end{cases},$$

fica:

$$\max \quad \sum_{i=1}^{nm} \sum_{j=1}^{nd} x_{ij} \cdot p_i \cdot n_{ij} + \sum_{i=1}^{nm} \sum_{j=1}^{nd} y_{ij} \cdot p_i \cdot n_{ij}^*$$

$$\text{s.a} \quad \sum_{j=1}^{nd} x_{ij} \le q_i \qquad i = 1 \ldots nm$$

$$\sum_{j=1}^{nd} y_{ij} \le r_i \qquad i = 1 \ldots nm$$

$$\sum_{i=1}^{nm} x_{ij} \ge d_j \qquad j = 1 \ldots nd$$

$$\sum_{i=1}^{nm} y_{ij} \ge e_j \qquad j = 1 \ldots nd$$

$$\sum_{k=1}^{nd_1} x_{i,j+k-1} + \sum_{k=1}^{nd_1} y_{i,j+k-1} \le 1 \qquad \begin{array}{l} i = 1, \ldots, nm \\ j = 1, \ldots, nd - nd_1 + 1 \end{array}$$

$$x_{i,S} + x_{i,S+7} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + y_{i,S} + y_{i,S+7} + y_{i,S+14} + y_{i,S+21} + y_{i,S+28} \le nsab$$

$$\begin{cases} x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} + \\ y_{i,S-6} + y_{i,S+1} + y_{i,S+8} + y_{i,S+15} + y_{i,S+22} + y_{i,S+29} \le ndom \end{cases}$$

$$\begin{cases} x_{i,S} + x_{i,S+7} + x_{i,S+14} + x_{i,S+21} + x_{i,S+28} + x_{i,S-6} + x_{i,S+1} + x_{i,S+8} + x_{i,S+15} + x_{i,S+22} + x_{i,S+29} + \\ y_{i,S} + y_{i,S+7} + y_{i,S+14} + y_{i,S+21} + y_{i,S+28} + y_{i,S-6} + y_{i,S+1} + y_{i,S+8} + y_{i,S+15} + y_{i,S+22} + y_{i,S+29} \le nfim \end{cases}$$

$$x_{i,S-6} + x_{i,S} + x_{i,S+1} + y_{i,S-6} + y_{i,S} + y_{i,S+1} \le 1$$

$$x_{i,S} + x_{i,S+1} + x_{i,S+7} + x_{i,S+8} + y_{i,S} + y_{i,S+1} + y_{i,S+7} + y_{i,S+8} \le 1$$

$$\vdots$$

$$x_{i,S+21} + x_{i,S+22} + x_{i,S+28} + x_{i,S+29} + y_{i,S+21} + y_{i,S+22} + y_{i,S+28} + y_{i,S+29} \le 1$$

$$\sum_{j=1}^{nd_1-1-udt} x_{ij} + \sum_{j=1}^{nd_1-1-udt} y_{ij} = 0 \quad i = 1,\ldots,nm, \text{ se } udt < nd_1 - 1$$

$x_{i,S} + x_{i,S+1} + y_{i,S} + y_{i,S+1} = 0$ se o militar i trabalhou no plantão no último final de semana do mês anterior

$x_{ij} \geq 0$, binárias

$y_{ij} \geq 0$, binárias

Tem-se $2nd \cdot nm$ variáveis, e um número de restrições no mínimo igual a: $2nm + 2nd + (nd - nd_1 + 1) \cdot nm + 3nm \cdot$ (número de finais de semana).

No próximo capítulo poderá ser observada a implementação computacional utilizando o *LINGO* e os resultados obtidos com exemplos no anexo 1.

CAPÍTULO VII

7. IMPLEMENTAÇÃO COMPUTACIONAL E AVALIAÇÃO DOS RESULTADOS

7.1 IMPLEMENTAÇÃO COMPUTACIONAL

Para fazer com que a criação do arquivo (LTF), citado no Capítulo V, item 5.2.2, a ser resolvido pelo *LINGO* ficasse mais fácil, foi necessário o desenvolvimento de um programa que executasse a interface entre os dados de entrada e o *LINGO*, e que analisasse a saída fornecida pelo *LINGO* e a transformasse em uma resposta mais compreensível pelo usuário final do sistema.

Este programa foi desenvolvido em *Visual Basic* 6.0 (*VB6*), e para que seu funcionamento fosse o mais amplo possível, optou-se em trabalhar de forma que o mesmo pudesse interpretar comandos salvos em um arquivo texto. Tal escolha visou possibilitar ao usuário uma maior flexibilidade na entrada de dados.

Para se ter acesso ao programa, o usuário deve entrar no executável Gerador. Ao entrar, aparecerá a tela da figura 7.1 a seguir:

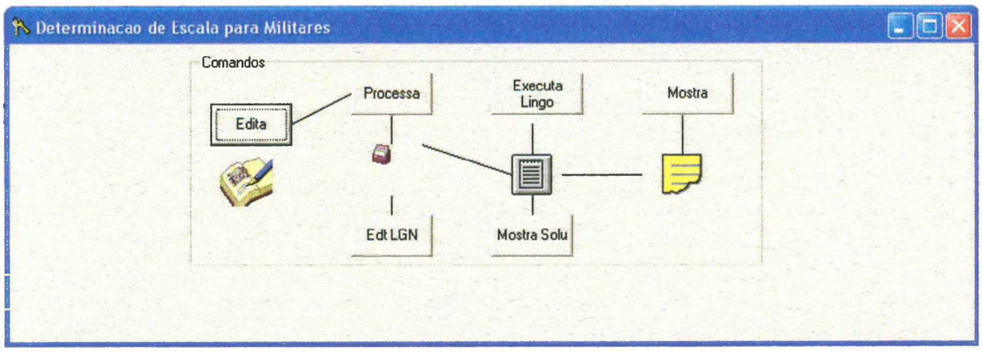


FIGURA 7.1 – JANELA PRINCIPAL DO PROGRAMA GERADOR

Para informar os dados do período, ou fazer as modificações necessárias para um novo período, o usuário deverá pressionar o botão 

, que chamará o *WORDPAD*, e aparecerá a seguinte janela da figura 7.2:

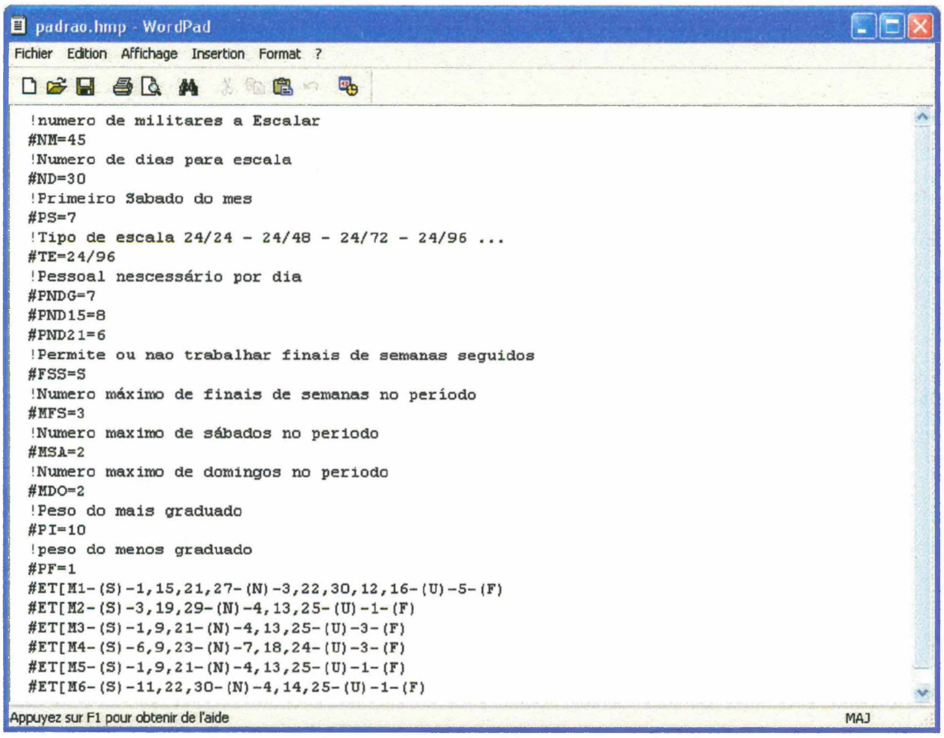


FIGURA 7.2 – JANELA WORDPAD/EDITA

Todos os comandos desse programa devem seguir a sintaxe abaixo:

- **#NM=** Indica a quantidade de militares do problema. Após o sinal de igual deve ser informada a quantidade.
- **#ND=** Indica o número de dias do problema, que devem ser informados após a igualdade.
- **#PS=** Indica o primeiro sábado do mês. Este parâmetro é utilizado para determinar os finais de semana do período de trabalho.
- **#TE=** Indica o Tipo de Escala, e deve ser informado após a igualdade. Os valores válidos são: 24/24, 24/48, 24/72, 24/96 ou 24/120 que indicam, respectivamente, 1 dia trabalhado por 1 de folga, 1 para 2, um para 3 um para 4 ou 1 para 5.
- **#PNDG=** Indica a quantidade de militares necessária por dia. Este comando indica um único valor a ser colocado para todos os dias. Em situações

especiais onde uma quantidade diferenciada seja necessária em um dia específico, pode ser usado o comando a seguir.

- **#PND?=** Onde no lugar da interrogação deve ser colocado um dia específico variando de 1 a ND, esta carga será considerada somente para este dia, e este comando deve vir após o comando #PNDG.
- **#PDSA=** Indica a quantidade diária de militares a serem colocados de sobreaviso, caso este comando não exista ou a quantidade for 0 (zero) o problema será resolvido segundo a formulação indicada em 6.2.5; caso o valor seja diferente de zero o problema será resolvido conforme a formulação indicada em 6.3.5.
- **#CE[M?-** Carga especial para o militar "?", quando um determinado militar deve ter uma carga especificada. Se, por exemplo, o militar 5 trabalha três dias no período no plantão, então **#CE[M5-3.**
- **#FSS=** Aceita como parâmetros, após a igualdade, S ou N, indicando se é permitido (S) ou não (N) que os militares trabalhem finais de semana seguidos.
- **#MFS=** Quantidade máxima de finais de semana que podem ser trabalhados por militar.
- **#MSA=** Quantidade máxima de sábados que podem ser trabalhados por mês por militar.
- **#MDO=** Quantidade máxima de domingos que podem ser trabalhadas por militar.
- **#PI=** Peso (nota) a ser atribuído ao militar mais graduado.
- **#PF=** Peso (nota) a ser atribuído ao militar menos graduado.

O programa estabelecerá pesos linearmente distribuídos entre os demais militares por ordem de antiguidade.

- **#ET[M?-(S)-s1,s2...-(N)-n1,n2...-(U)-u-(F) -f** onde:

?	indica o número do militar;
s1,s2...	indicam os dias de escolha preferencial para trabalho deste militar;
n1,n2...	indicam os dias para os quais este militar não gostaria de trabalhar;
u	indica a quantos dias o militar fez seu último plantão no período anterior;
f	1 indica que trabalhou no último final de semana do mês anterior, 0 caso contrário.

Exemplificando, pode-se ter: #ET[M4-(S)-2,7,21-(N)-5,9,17-(U)-2-(F)-1, ou seja, o militar 4 escolheu os dias 2, 7 e 21 como preferenciais para trabalho no plantão, preferindo não trabalhar no plantão nos dias 5, 9 e 17 e trabalhou a dois dias do início do período de escala, ou seja, considerando um mês normal de 30 dias, trabalhou no dia 29. Trabalhou no último final do mês anterior.

- **#ZV[M?-z1,z2,...** onde:

?	indica o número do militar;
z1,z2...	são variáveis que determinam o intervalo onde assumirão valor zero, neste caso x(?,z1)=0, x(?,z1+1);...; x(?,z1+1)=0. que são necessários, por exemplo, quando um militar entra em férias num determinado período.

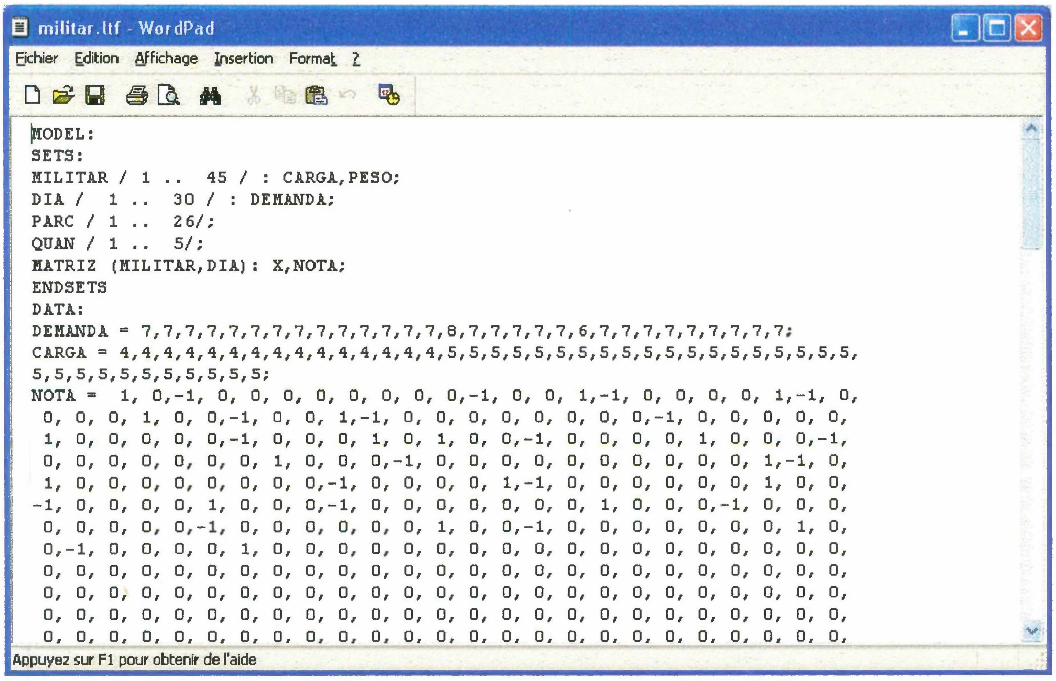
- **#SV[M?-s1,s2,...onde**

?	indica o número do militar;
s1,s2...	são variáveis que assumiram valor 1 (um), neste caso x(?,s1)=1, x(?,s2)=1.

Depois de informar os dados, deve-se salvar e fechar o programa, voltando à janela principal do Programa Gerador.

Então, pressionar o botão 

que fará o programa ler as informações salvas no formato texto e construir o modelo que seja compreensível pelo *LINGO* e que depois de resolvido pelo mesmo, forneça a resposta otimizada para o problema em questão. Este programa pode ser visualizado como na figura 7.3, pressionando-se a tecla 




```
MODEL:
SETS:
MILITAR / 1 ..  45 / : CARGA,PESO;
DIA /  1 ..  30 / : DEMANDA;
PARC / 1 ..  26/;
QUAN / 1 ..  5/;
MATRIZ (MILITAR,DIA): X,NOTA;
ENDSETS
DATA:
DEMANDA = 7,7,7,7,7,7,7,7,7,7,7,7,7,7,8,7,7,7,7,7,6,7,7,7,7,7,7,7,7,7;
CARGA = 4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
5,5,5,5,5,5,5,5,5,5,5;
NOTA =  1, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 1,-1, 0, 0, 0, 0, 1,-1, 0,
 0, 0, 0, 1, 0, 0,-1, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 1, 0, 1, 0, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0,-1,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0,
 1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
-1, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0, 0, 0,
 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
 0,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

FIGURA 7.3 – JANELA WORDPAD/EDT LGN

Esta tela não possui utilidade para o usuário, servindo apenas para avaliar a forma de um texto que será resolvido pelo *LINGO* em seu formato ".LTF."

Pressionando-se a tecla 

o programa chamará o *LINGO*, que fará os cálculos e encontrará a solução para o problema proposto. Ao pressionar esta tecla, aparecerá a janela do *LINGO* da figura 7.4 a seguir:

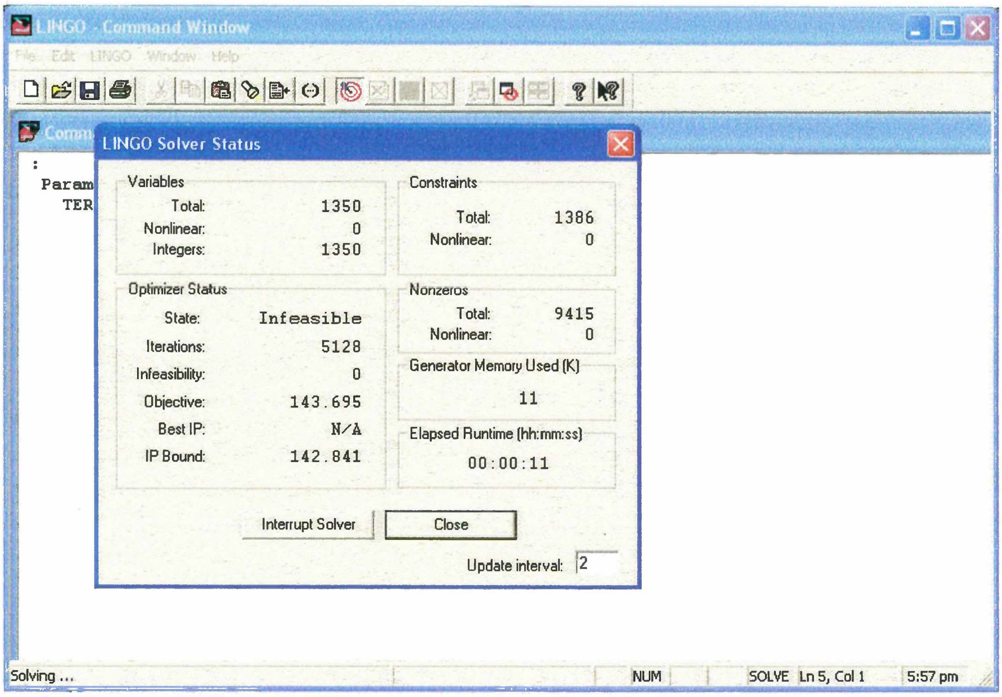
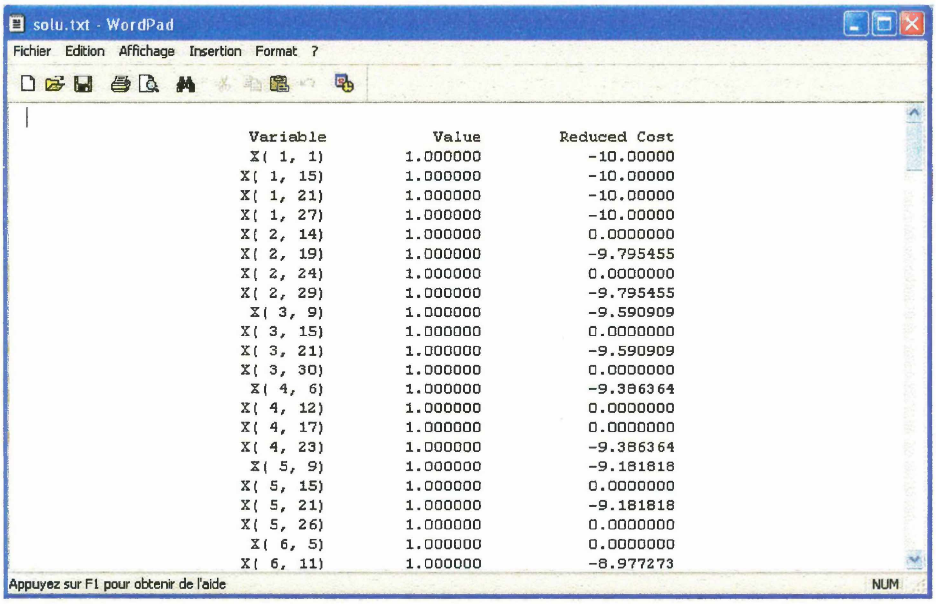


FIGURA 7.4 – JANELA *LINGO*

Assim que os cálculos são completados, o *LINGO* fecha e a janela Principal do Programa Gerador retorna.

A tecla 

ao ser pressionada, mostra a solução encontrada pelo *LINGO*, mas que para o usuário não tem bom formato para análise de respostas, como se pode observar na figura 7.5:



Variable	Value	Reduced Cost
X(1, 1)	1.000000	-10.00000
X(1, 15)	1.000000	-10.00000
X(1, 21)	1.000000	-10.00000
X(1, 27)	1.000000	-10.00000
X(2, 14)	1.000000	0.0000000
X(2, 19)	1.000000	-9.795455
X(2, 24)	1.000000	0.0000000
X(2, 29)	1.000000	-9.795455
X(3, 9)	1.000000	-9.590909
X(3, 15)	1.000000	0.0000000
X(3, 21)	1.000000	-9.590909
X(3, 30)	1.000000	0.0000000
X(4, 6)	1.000000	-9.386364
X(4, 12)	1.000000	0.0000000
X(4, 17)	1.000000	0.0000000
X(4, 23)	1.000000	-9.386364
X(5, 9)	1.000000	-9.181818
X(5, 15)	1.000000	0.0000000
X(5, 21)	1.000000	-9.181818
X(5, 26)	1.000000	0.0000000
X(6, 5)	1.000000	0.0000000
X(6, 11)	1.000000	-8.977273

FIGURA 7.5 – JANELA WORDPAD/MOSTRA SOLUÇÃO

Para uma boa visualização pelo usuário, basta que a tecla Mostra seja pressionada. Ele encontrará as respostas como se observa na figura 7.6 a seguir:

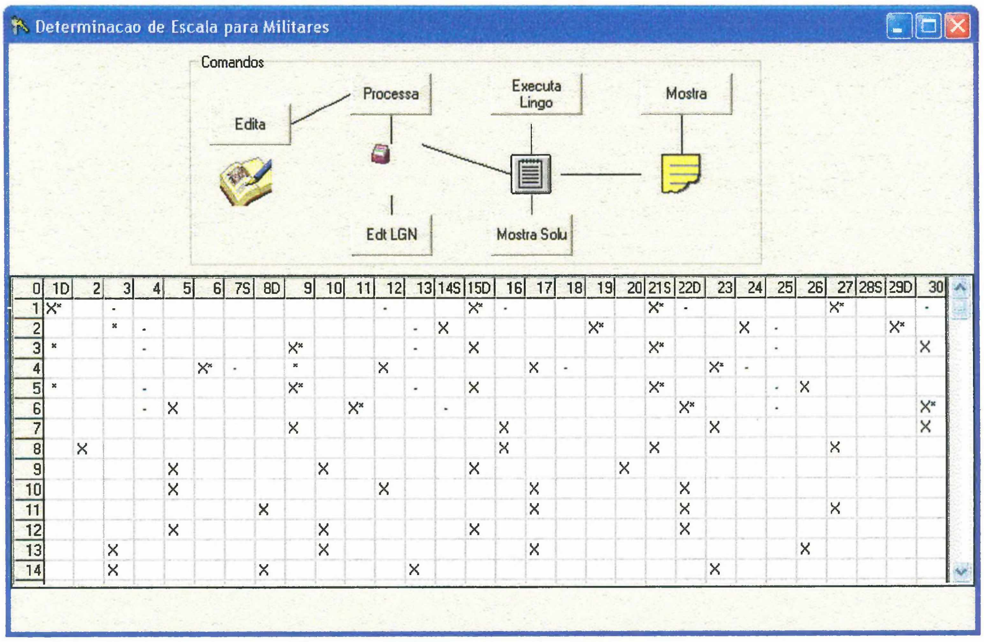


FIGURA 7.6 – JANELA MOSTRA

7.2 AVALIAÇÃO DOS RESULTADOS

Para avaliação dos resultados, algumas simulações foram efetuadas, alterando-se quantidade de militares, demandas diárias, escalas e outros dados.

Para parte destas avaliações foi pedido para que os militares preenchessem uma lista indicando 3 dias nos quais prefeririam trabalhar no plantão e com 6 dias no qual sua escolha seria não trabalhar no plantão. Para a escolha dos dias de trabalho, a única condição imposta é que eles tivessem entre si uma diferença de no mínimo 5 dias. Nenhuma restrição foi imposta para os dias que eles escolheriam para não trabalhar.

Em outros exemplos, foram realizadas simulações usando um programa para este fim.

Todos os tempos aqui indicados foram obtidos rodando-se o *LINGO 6.0* em um computador Pentium 4, 1.4 GHz, 384 Mb de Memória RAM e HD de 40 Gb.

As escolhas diárias para cada militar podem ser observadas no anexo 1.

Para os exemplos citados neste capítulo, serão considerados:

FSS – permite ou não o trabalho em finais de semanas seguidos (S/N);

Erro 1 – informa as não indicações para trabalho no plantão em dias escolhidos como preferenciais pelos militares;

Erro 2 – informa a quantidade de indicações para trabalho no plantão em dias escolhidos para não serem trabalhados pelos militares;

Atend.1 – informa o percentual de atendimento dos dias escolhidos para plantão;

Atend 2 – informa o percentual de não indicação para dias escolhidos para folga do plantão.

7.2.1 Exemplos com Alteração da Quantidade de Militares e Escala – sem Sobreaviso

A seguir, são apresentados três exemplos numéricos para um melhor entendimento do procedimento até aqui exposto.

Exemplo 1(dados levantados junto aos militares):

quantidade de militares: 90;

quantidade de dias para escala: 30;

demanda diária de 19 militares;

máximo de finais de semana trabalhados no período: 2;

total de variáveis binárias do problema: 2700;

total de escolhas para trabalhar no plantão: 3x90 = 270;

total de escolhas para não trabalhar no plantão = 6*90 = 540.

As escolhas dos militares podem ser encontrados no Anexo 1 do trabalho. A Tabela 7.1 mostra uma análise dos resultados obtidos para o exemplo 1.

TABELA 7.1 – ANÁLISE DE RESULTADOS – EXEMPLO 1

Escala	FSS	Tempo (min)	Erro 1 (quantidade)	Erro 2 (quantidade)	Atend. 1 (%)	Atend. 2 (%)
24/24	S	0:01.28	4	0	98,52	100
	N	0:01.71	12	0	95,56	100
24/48	S	0:03.93	5	0	98,15	100
	N	0:09.00	13	0	95,19	100
24/72	S	0:17.33	30	6	88,89	98,89
	N	7:11.75	42	11	84,44	97,96
24/96	S	0:14.84	Infactível	Infactível	Infactível	Infactível
	N	0:09.03	Infactível	Infactível	Infactível	Infactível

Exemplo 2 (simulação):

quantidade de militares: 65;

quantidade de dias para escala: 30;

demanda diária: 9 militares;

máximo de finais de semana trabalhados no plantão no período: 2;

total de variáveis binárias do problema: 1950;

total de escolhas para trabalhar no plantão: 3x65 = 195;

total de escolhas para não trabalhar no plantão = 6*65 = 390.

A análise dos resultados do exemplo 2 encontra-se na tabela 7.2 a seguir:

TABELA 7.2 – ANÁLISE DE RESULTADOS – EXEMPLO 2

Escala	FSS	Tempo (min)	Erro 1 (quantidade)	Erro 2 (quantidade)	Atend. 1 (%)	Atend. 2 (%)
24/24	S	0:00.86	4	0	97,95	100
	N	0:00.89	8	0	95,90	100
24/48	S	0:00.95	5	0	97,44	100
	N	0:00.99	9	0	95,38	100
24/72	S	0:01.00	9	0	95,38	100
	N	0:01.64	13	0	93,33	100
24/96	S	0:09.30		Infactível		
	N	0:04.49		Infactível		

Exemplo 3 (simulação): Grande Quantidade de Militares

Um exemplo maior, com 1000 militares e demanda diária de 212 militares para 30 dias, foi gerado a partir de um programa criado para isto. Neste exemplo foi permitido o trabalho em finais de semanas seguidos, e foram aceitos no máximo dois trabalhos em finais de semana no período. A escala indicada foi de 24/72 horas.

Após 1:45h de processamento o *LINGO* indicava a factibilidade, porém ainda não tinha sido alcançada a otimalidade, neste instante a resolução foi interrompida. A resposta obtida indicava um erro do tipo 1 de 64 unidades e nenhum erro do tipo 2. Isto significa um índice de atendimento do tipo 1 de 97,87% e um índice do tipo 2 de 100%.

A seguir, apresenta-se um exemplo (simulação) considerando a idéia do militar de sobreaviso.

7.2.2 Exemplo com Alteração da Escala e da Demanda - Sobreaviso

Considerando agora 75 militares, não podendo trabalhar finais de semana consecutivos, sendo possível trabalhar no máximo dois finais de semana no mês de 30 dias, e considerando o peso do mais graduado 10 e do menos graduado 1. Com a alteração do tipo de escala e da quantidade necessária de militares no dia, obtiveram-se os resultados apresentados na tabela 7.3 a seguir:

TABELA 7.3 – RESULTADOS PARA MUDANÇA DE DEMANDA E ESCALA

DEMANDA	ESCALA 24/96			ESCALA 24/72			ESCALA 24/48			ESCALA 24/24		
	Tempo	Erro 1	Erro 2	Tempo	Erro 1	Erro 2	Tempo	Erro 1	Erro 2	Tempo	Erro 1	Erro 2
8	0:03.00	38	0	0:01.64	30	0	0:01.28	23	0	0:00.97	23	0
9	0:03.65	30	0	0:01.87	23	0	0:01.25	16	0	0:01.00	16	0
10	0:07.19	27	1	0:01.68	20	0	0:01.18	13	0	0:01.01	13	0
11	1:50.00	39	1	0:01.71	19	0	0:01.34	12	0	0:01.98	12	0
12	3:18.00	56	3	0:07.36	19	0	0:01.96	12	0	0:01.33	12	0
13	0:03.00	infactível		0:10.08	20	1	0:01.48	12	0	0:01.04	12	0
14				0:17.42	20	2	0:04.31	12	0	0:01.14	12	0
15				0:18.80	20	5	0:06.01	12	0	0:01.17	12	0
16				4:16.60	62	12	0:13.04	13	0	0:01.75	12	0
17				12:00.00	infactível		0:06.46	15	1	0:01.36	12	0
18				0:15.47	infactível		1:06.74	24	6	0:04.00	13	0
19							0:08.59	infactível		0:06.69	infactível	

7.2.3 Exemplo com Alteração da Quantidade de Escolhas

Considerando agora 75 militares, com demanda diária de 8 militares não podendo trabalhar no plantão finais de semana consecutivos, sendo possível trabalhar no plantão no máximo dois finais de semana no mês de 30 dias, e considerando o peso do mais graduado 10 e do menos graduado 1, numa escala 24/96. Com a alteração da quantidade de escolhas para os dias em que o militar deseja trabalhar no plantão, obteve-se os seguintes resultados apresentados na tabela 7.4:

TABELA 7.4 – ANÁLISE DE RESULTADOS – MUDANÇA DA QUANTIDADE DE ESCOLHA

Escolha de Sim	Escolha de Não	FSS	Tempo (mm:ss:dd)	Erro 1 (quantidade)	Erro 2 (quantidade)	Atend. 1 (%)	Atend. 2 (%)
3	6	N	0:02.58	23	0	89,8	100
4	6	N	0:06.24	62	0	79,3	100
5	6	N	0:02.10	91	0	75,7	100
5	10	N	0:16.51	103	0	72,5	100
5	15	N	0:08.42	103	30	72,5	97,3
5	20	N	0:02.66	108	19	71,2	98,7
6	20	N	0:06.58	161	17	64,2	98,9

7.2.4 Exemplos com Alteração de Pesos

Com a finalidade de avaliar o impacto que a variação de pesos causa nos erros foram simulados exemplos abaixo. Em todas as avaliações considerou-se 90 militares escolhendo 5 dias para trabalhar e 15 dias para não trabalhar. A grande quantidade de escolhas é para que ocorram muitos erros, possibilitando uma melhor visualização das diferenças.

A análise de um exemplo com demanda diária de 20 militares com escala de 24/48 é apresentada na tabela 7.5 a seguir:

TABELA 7.5 – ANÁLISE DE PESOS COM DEMANDA 20/DIA – ESCALA 24/48

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	0:02.09	80	3	83			
1	10	0:02.95	82	7	89	2,5	133	7,22
1	120	0:18.85	87	9	96	8,75	200	15,66
1	600	0:03.37	85	11	96	6,25	266	15,66

A mesma análise foi efetuada considerando a necessidade de 2 militares de sobreaviso e demanda diária de 18 militares (total 20 militares), com resultados apresentados na tabela 7.6:

TABELA 7.6– ANÁLISE DE PESOS COM DEMANDA (18+2)/DIA - ESCALA 24/48

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	1:17:44	81	2	83			
1	10	0:21.43	82	7	89	1,23	250	7,22
1	120	0:27.09	85	11	96	4,94	450	15,66
1	600	0:27.08	85	11	96	4,94	450	15,66

As mesmas análises são feitas para a escala de 24/72 , cujos resultados são apresentados nas tabelas 7.7 e 7.8 a seguir:

TABELA 7.7 – ANÁLISE DE PESOS COM DEMANDA 20/DIA - ESCALA 24/72

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	0:17.32	139	9	148			
1	10	0:07.48	142	13	155	2,16	44,44	4,73
1	120	0:08.52	144	13	157	3,60	44,44	6,08
1	600	0:16.39	145	12	157	4,32	33,33	6,08

TABELA 7.8 – ANÁLISE DE PESOS COM DEMANDA (18+2)/DIA – ESCALA 24/72

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	1:52.73	141	7	148			
1	10	3:35.51	141	14	155	0	100	4,73
1	120	5:52.56	145	12	157	2,84	71,43	6,08
1	600	11:01.69	145	12	157	2,84	71,43	6,08

Uma simulação considerando uma demanda de 21 militares sem sobreaviso e uma demanda de 19militares com 2 de sobreaviso (total de 21), tem resultados apresentados nas tabelas 7.9 e 7.10 a seguir:

TABELA 7.9 – ANÁLISE DE PESOS COM DEMANDA 21/DIA – ESCALA 24/72

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	0:18.15	147	20	167			
1	10	0:10.28	149	25	174	1,36	25	4,19
1	120	0:13.37	149	31	180	1,36	55	7,78
1	600	0:11.10	149	31	180	1,36	55	7,78

TABELA 7.10– ANÁLISE DE PESOS COM DEMANDA (19+2)/DIA – ESCALA 24/72

Menor Peso	Maior Peso	Tempo (mm:ss.dd)	Erro1 (quant)	Erro2 (quant)	Erro Total (quant)	Aumento erro 1 (%)	Aumento erro 2 (%)	Aumento total (%)
1	1	2:30.78	148	19	167			
1	10	1:00.14	149	25	174	6,76	31,58	4,19
1	120	3:31.25	149	31	180	6,76	63,16	7,78
1	600	4:13.70	149	31	180	6,76	63,16	7,78

CAPÍTULO VIII

8. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

8.1 CONCLUSÕES

Dos vários testes executados, considerando diferentes quantidades de militares e diferentes demandas diárias, observou-se a validade da formulação através de aplicações práticas.

Foram efetuadas variações nos pesos atribuídos à antiguidade de cada militar, e foi observado que a diferença de pesos é significativa somente quando a quantidade de dias para escolha de cada militar é próxima da demanda necessária deste militar. Analisando os resultados optou-se por considerar o menor peso igual a 1 e maior peso igual a 10, pois dá-se desta maneira uma vantagem para os militares mais antigos, porém sem onerar excessivamente os mais novos.

A idéia do sobreaviso mostrou-se extremamente interessante para o gerenciamento do pessoal, evitando atropelos de chamada de novos militares no instante de faltas.

Quando o número de militares disponíveis é muito pequeno, porém, a idéia de sobreaviso torna-se inviável, pois não existe folga suficiente para o atendimento da demanda criada. Neste caso a resolução é feita pelo modelo comum, e no caso de faltas o modelo é executado novamente para o período restante, no entanto os resultados apresentados (movimentação de muitos militares para outros dias diferentes dos originalmente indicados) não foi animador.

A utilização do *LINGO* para a resolução de PLI mostrou-se muito eficiente e a chamada por programas externos é muito fácil. É uma excelente ferramenta para desenvolvimento de soluções de uma maneira rápida.

8.2 SUGESTÕES PARA TRABALHOS FUTUROS

Uma possibilidade de continuação para o presente trabalho seria o desenvolvimento de um programa computacional específico para o problema de Programação Inteira Binária aplicável a este tipo de problema. Com isto não seria necessária a utilização do *LINGO*, que apesar de extremamente eficiente, tem um custo elevado.

Uma idéia de desenvolvimento seria a de trabalhar com um algoritmo baseado em Balas, aproveitando sua característica de gerar todas as soluções, com isto ter-se-ia uma maior gama de soluções para apresentar ao encarregado pela confecção das escalas.

Também existe a possibilidade de estudos de aplicações das formulações apresentadas em outros problemas que apresentem características semelhantes, como funcionários de usinas e refinarias, dentre outros.

ANEXO 1 – EXEMPLO COMPLETO DE APLICAÇÃO DO PROGRAMA PARA ESCALA DE SERVIÇO DE PLANTÃO

Abaixo o conteúdo do arquivo de entrada do programa GERADOR, onde se encontram as escolhas de 90 militares para os dias em que gostaria de trabalhar e os que não gostaria de trabalhar e quantidade de dias desde o último dia trabalhado no mês anterior, além dos demais parâmetros de solução.

Tomando-se como exemplo o Militar 1 (M1), verifica-se que escolheu para trabalhar os dias 6, 12 e 23 e para não trabalhar os dias 3, 5, 7, 17, 20 e 22, e que o último dia em que trabalhou no mês anterior foi há dois dias antes do início da nova escala.

```
!Número de militares
#NM=90
!Número de dias para escala
#ND=30
!Primeiro sábado do mês
#PS=6
!Tipo de escala 24/24 - 24/48 - 24/72 - 24/96 ...
#TE=24/72
!Pessoal nescessário por dia
#PNDG=19
!Permite ou não trabalhar finais de semanas seguidos
#FSS=N
!Número máximo de finais de semanas no período
#MFS=2
!Número máximo de sábados no período
#MSA=2
!Número máximo de domingos no período
#MDO=2
!Peso do mais graduado
#PI=10
!Peso do menos graduado
#PF=1
#ET[M1-(S)- 23, 6, 12-(N)- 7, 3, 5, 17, 22, 20-(U)- 2-(F)
#ET[M2-(S)- 4, 15, 24-(N)- 22, 23, 3, 18, 12, 11-(U)- 1-(F)
#ET[M3-(S)- 19, 10, 26-(N)- 25, 9, 17, 23, 22, 8-(U)- 4-(F)
#ET[M4-(S)- 27, 2, 21-(N)- 18, 13, 29, 26, 8, 3-(U)- 2-(F)
#ET[M5-(S)- 8, 16, 24-(N)- 23, 4, 9, 30, 15, 20-(U)- 4-(F)
#ET[M6-(S)- 4, 24, 19-(N)- 10, 14, 3, 16, 27, 6-(U)- 1-(F)
#ET[M7-(S)- 10, 2, 27-(N)- 15, 3, 8, 13, 20, 18-(U)- 4-(F)
#ET[M8-(S)- 23, 30, 18-(N)- 14, 27, 10, 15, 8, 25-(U)- 3-(F)
#ET[M9-(S)- 18, 25, 1-(N)- 24, 4, 28, 8, 5, 22-(U)- 3-(F)
#ET[M10-(S)- 14, 30, 20-(N)- 19, 7, 27, 16, 22, 5-(U)- 4-(F)
#ET[M11-(S)- 26, 1, 6-(N)- 9, 14, 10, 25, 13, 8-(U)- 2-(F)
#ET[M12-(S)- 23, 7, 13-(N)- 22, 28, 8, 6, 19, 3-(U)- 4-(F)
#ET[M13-(S)- 3, 23, 18-(N)- 11, 4, 13, 20, 26, 27-(U)- 3-(F)
#ET[M14-(S)- 12, 26, 20-(N)- 13, 16, 7, 19, 28, 29-(U)- 2-(F)
#ET[M15-(S)- 10, 21, 3-(N)- 16, 20, 28, 12, 26, 27-(U)- 3-(F)
#ET[M16-(S)- 18, 26, 6-(N)- 17, 23, 21, 20, 9, 25-(U)- 5-(F)
#ET[M17-(S)- 18, 13, 7-(N)- 12, 29, 19, 14, 26, 24-(U)- 4-(F)
#ET[M18-(S)- 6, 22, 15-(N)- 24, 27, 18, 16, 21, 9-(U)- 3-(F)
```

#ET[M19-(S)- 18, 29, 5-(N)- 7, 28, 19, 12, 10, 6-(U)- 4-(F)
#ET[M20-(S)- 30, 17, 22-(N)- 26, 23, 20, 7, 18, 19-(U)- 2-(F)
#ET[M21-(S)- 22, 5, 16-(N)- 18, 23, 28, 3, 21, 10-(U)- 4-(F)
#ET[M22-(S)- 30, 5, 15-(N)- 20, 29, 10, 14, 4, 23-(U)- 3-(F)
#ET[M23-(S)- 5, 19, 25-(N)- 16, 29, 28, 30, 22, 24-(U)- 2-(F)
#ET[M24-(S)- 13, 4, 28-(N)- 16, 12, 3, 21, 14, 11-(U)- 2-(F)
#ET[M25-(S)- 11, 28, 4-(N)- 10, 23, 18, 14, 21, 27-(U)- 5-(F)
#ET[M26-(S)- 11, 19, 26-(N)- 6, 7, 14, 10, 24, 25-(U)- 4-(F)
#ET[M27-(S)- 9, 15, 30-(N)- 12, 18, 20, 6, 5, 7-(U)- 4-(F)
#ET[M28-(S)- 8, 28, 13-(N)- 6, 27, 25, 29, 22, 26-(U)- 4-(F)
#ET[M29-(S)- 13, 22, 8-(N)- 26, 4, 15, 24, 17, 10-(U)- 2-(F)
#ET[M30-(S)- 10, 25, 19-(N)- 27, 22, 11, 26, 12, 3-(U)- 2-(F)
#ET[M31-(S)- 12, 24, 3-(N)- 6, 21, 13, 20, 4, 29-(U)- 1-(F)
#ET[M32-(S)- 12, 7, 26-(N)- 21, 10, 27, 29, 13, 15-(U)- 4-(F)
#ET[M33-(S)- 28, 4, 13-(N)- 8, 12, 15, 16, 29, 23-(U)- 1-(F)
#ET[M34-(S)- 19, 2, 11-(N)- 27, 5, 26, 22, 17, 20-(U)- 5-(F)
#ET[M35-(S)- 5, 22, 10-(N)- 8, 4, 13, 25, 20, 6-(U)- 1-(F)
#ET[M36-(S)- 23, 16, 5-(N)- 17, 8, 14, 11, 7, 28-(U)- 4-(F)
#ET[M37-(S)- 7, 17, 23-(N)- 11, 14, 18, 4, 16, 24-(U)- 5-(F)
#ET[M38-(S)- 18, 11, 3-(N)- 17, 16, 4, 23, 8, 26-(U)- 2-(F)
#ET[M39-(S)- 14, 8, 28-(N)- 29, 27, 20, 24, 9, 15-(U)- 3-(F)
#ET[M40-(S)- 27, 3, 22-(N)- 18, 7, 9, 17, 19, 10-(U)- 2-(F)
#ET[M41-(S)- 17, 12, 6-(N)- 13, 19, 4, 23, 7, 22-(U)- 4-(F)
#ET[M42-(S)- 3, 27, 8-(N)- 22, 18, 23, 20, 26, 15-(U)- 5-(F)
#ET[M43-(S)- 24, 10, 2-(N)- 8, 3, 11, 16, 21, 6-(U)- 2-(F)
#ET[M44-(S)- 8, 23, 18-(N)- 11, 22, 28, 25, 6, 27-(U)- 4-(F)
#ET[M45-(S)- 24, 12, 7-(N)- 11, 23, 28, 3, 26, 20-(U)- 2-(F)
#ET[M46-(S)- 22, 6, 16-(N)- 19, 7, 10, 15, 28, 13-(U)- 4-(F)
#ET[M47-(S)- 14, 27, 7-(N)- 3, 17, 25, 6, 8, 10-(U)- 1-(F)
#ET[M48-(S)- 29, 7, 12-(N)- 5, 10, 26, 23, 8, 17-(U)- 4-(F)
#ET[M49-(S)- 4, 22, 17-(N)- 13, 7, 24, 20, 16, 6-(U)- 2-(F)
#ET[M50-(S)- 1, 27, 17-(N)- 29, 25, 8, 28, 3, 7-(U)- 5-(F)
#ET[M51-(S)- 9, 21, 16-(N)- 19, 30, 7, 6, 12, 15-(U)- 3-(F)
#ET[M52-(S)- 19, 28, 10-(N)- 18, 17, 25, 6, 22, 9-(U)- 2-(F)
#ET[M53-(S)- 3, 12, 21-(N)- 7, 9, 27, 10, 20, 14-(U)- 3-(F)
#ET[M54-(S)- 22, 1, 14-(N)- 26, 29, 7, 24, 8, 28-(U)- 4-(F)
#ET[M55-(S)- 25, 4, 11-(N)- 29, 22, 26, 10, 12, 9-(U)- 3-(F)
#ET[M56-(S)- 28, 6, 14-(N)- 3, 18, 20, 21, 17, 11-(U)- 5-(F)
#ET[M57-(S)- 18, 13, 1-(N)- 15, 17, 4, 14, 5, 28-(U)- 1-(F)
#ET[M58-(S)- 10, 26, 15-(N)- 22, 27, 11, 28, 23, 19-(U)- 3-(F)
#ET[M59-(S)- 16, 3, 29-(N)- 25, 20, 5, 18, 28, 14-(U)- 3-(F)
#ET[M60-(S)- 22, 13, 3-(N)- 5, 21, 11, 27, 24, 26-(U)- 2-(F)
#ET[M61-(S)- 6, 19, 24-(N)- 8, 11, 23, 16, 9, 13-(U)- 5-(F)
#ET[M62-(S)- 25, 16, 5-(N)- 11, 4, 28, 14, 15, 12-(U)- 3-(F)
#ET[M63-(S)- 9, 17, 23-(N)- 25, 28, 3, 16, 29, 14-(U)- 3-(F)
#ET[M64-(S)- 23, 11, 5-(N)- 12, 18, 3, 15, 21, 26-(U)- 4-(F)
#ET[M65-(S)- 26, 12, 2-(N)- 24, 15, 20, 28, 13, 4-(U)- 4-(F)
#ET[M66-(S)- 20, 27, 11-(N)- 25, 15, 5, 12, 18, 23-(U)- 5-(F)
#ET[M67-(S)- 27, 7, 14-(N)- 4, 20, 12, 5, 13, 23-(U)- 2-(F)
#ET[M68-(S)- 10, 15, 27-(N)- 5, 18, 12, 14, 26, 28-(U)- 4-(F)
#ET[M69-(S)- 15, 23, 1-(N)- 11, 20, 25, 10, 19, 14-(U)- 5-(F)
#ET[M70-(S)- 22, 8, 16-(N)- 3, 18, 11, 6, 5, 19-(U)- 4-(F)
#ET[M71-(S)- 25, 17, 10-(N)- 20, 19, 16, 18, 15, 8-(U)- 4-(F)
#ET[M72-(S)- 3, 14, 9-(N)- 22, 28, 10, 27, 29, 6-(U)- 4-(F)
#ET[M73-(S)- 30, 6, 25-(N)- 28, 11, 8, 20, 21, 15-(U)- 4-(F)
#ET[M74-(S)- 20, 13, 26-(N)- 14, 21, 5, 9, 25, 29-(U)- 2-(F)
#ET[M75-(S)- 14, 7, 25-(N)- 10, 29, 13, 20, 15, 27-(U)- 2-(F)
#ET[M76-(S)- 11, 19, 3-(N)- 2, 21, 18, 15, 6, 22-(U)- 2-(F)
#ET[M77-(S)- 15, 23, 6-(N)- 13, 29, 22, 12, 21, 9-(U)- 2-(F)

```
#ET[M78-(S)- 24, 2, 15-(N)- 14, 20, 3, 21, 7, 18-(U)- 4-(F)
#ET[M79-(S)- 10, 25, 17-(N)- 19, 13, 22, 20, 26, 9-(U)- 4-(F)
#ET[M80-(S)- 17, 3, 11-(N)- 27, 26, 13, 25, 24, 23-(U)- 5-(F)
#ET[M81-(S)- 3, 19, 27-(N)- 4, 21, 22, 24, 10, 20-(U)- 2-(F)
#ET[M82-(S)- 29, 14, 19-(N)- 11, 18, 3, 16, 13, 10-(U)- 2-(F)
#ET[M83-(S)- 3, 21, 27-(N)- 20, 25, 4, 19, 9, 23-(U)- 4-(F)
#ET[M84-(S)- 12, 26, 18-(N)- 10, 9, 3, 23, 29, 5-(U)- 1-(F)
#ET[M85-(S)- 9, 16, 28-(N)- 26, 21, 5, 11, 15, 12-(U)- 4-(F)
#ET[M86-(S)- 11, 20, 3-(N)- 29, 14, 22, 12, 21, 17-(U)- 5-(F)
#ET[M87-(S)- 3, 26, 18-(N)- 23, 14, 24, 12, 5, 22-(U)- 1-(F)
#ET[M88-(S)- 25, 13, 30-(N)- 4, 21, 26, 16, 11, 24-(U)- 4-(F)
#ET[M89-(S)- 6, 23, 11-(N)- 14, 27, 24, 30, 21, 28-(U)- 4-(F)
#ET[M90-(S)- 23, 18, 12-(N)- 20, 17, 4, 5, 11, 19-(U)- 4-(F)
```

A seguir o conteúdo do arquivo de extensão "LTF" gerado pelo programa para ser resolvido pelo LINGO

```
MODEL:
SETS:
MILITAR / 1 .. 90 / : CARGA,PESO;
DIA / 1 .. 30 / : DEMANDA;
PARC / 1 .. 27/;
QUAN / 1 .. 4/;
MATRIZ (MILITAR,DIA): X,NOTA;
ENDSETS
DATA:
DEMANDA = 19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,19,
19,19,19,19,19,19,19,19;
CARGA = 6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,
7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7;
NOTA =  0, 0,-1, 0,-1, 1,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0,-1, 0, 0,-1, 0,-1, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0, 0, 0,-1,-1, 0, 0, 1, 0, 0,-1,
 0, 0, 0,-1,-1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1,-1, 1, 0, 0, 0,
 0, 0, 0,-1, 0, 1, 0, 0,-1,-1, 0,-1, 1, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0,-1,
 0, 0, 0, 0,-1, 0, 0, 0, 0,-1, 0, 0, 1, 0, 0, 0, 0,-1, 1, 0,-1, 0, 0, 0, 0,
-1, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0,-1, 0, 0,-1, 1, 0, 0, 0, 0,
 0,-1, 0, 0,-1, 1, 0,-1, 0, 0, 0,-1, 0, 0, 0,-1, 0,-1, 0, 0, 1, 0, 0, 0, 0,
 1, 0, 0,-1, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0,-1, 0, 1, 0, 0,-1, 0,-1, 0, 0,-1,
 0,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0,-1, 0, 0, 0,
-1,-1, 0, 0, 1, 0, 0, 0, 0, 1, 0,-1, 0,-1, 0, 0, 1, 1, 0, 0,-1,-1, 0, 0,-1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0,-1, 1, 0, 0,-1, 0, 0, 0, 0, 0,
 0,-1, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0,-1, 1, 0,-1, 0, 0, 0, 0,-1, 0,
 0, 1, 1, 0, 0, 0, 0, 1, 0,-1,-1,-1, 0, 0,-1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0,-1, 1, 0, 0, 0, 0, 0, 0,-1, 0, 0,-1, 1,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
-1, 0, 0,-1, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0,-1, 0,-1,
 0, 0, 0, 0, 1, 0,-1, 0, 0, 1, 0, 0,-1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0,
 0, 0, 0, 1,-1, 0, 0,-1, 0, 0,-1, 1, 0, 0, 0, 0, 0, 1, 0,-1,-1, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0, 0,-1, 0, 0, 0,-1, 1, 0, 0, 0, 0,-1,-1,-1,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0,-1, 1, 0,-1,-1, 0,-1,
 0,-1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,-1, 1,-1, 0, 0, 0, 1,
-1, 0, 0, 0, 0,-1, 0,-1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 0, 0,
 0, 1,-1, 0,-1, 0, 0,-1, 1, 0,-1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 1,-1,-1, 0,
 0,-1, 0,-1, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0,
 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1,-1,-1, 0, 1,-1, 0, 0,-1, 0, 0,
 0, 1, 0, 0,-1, 0, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0,-1, 1,-1,
 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0,-1, 0, 0, 0,-1, 1, 0, 0, 0,
 0,-1, 0, 0,-1, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0,-1, 0, 0, 1, 0, 0,-1, 0,-1, 1, 0, 0,-1,-1,-1, 0, 0,-1, 1, 0, 0, 0, 0,
```

```
 0, 0,-1,-1, 1,-1, 0,-1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0,-1, 1, 0, 0,-1, 0, 0, 0,-1, 0, 0,-1, 0,-1, 0, 0, 0,-1, 1,
 0, 0, 0, 0, 0, 0, 0,-1,-1, 0, 0,-1, 1, 0, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
-1,-1, 1, 0, 0, 0, 0, 0, 0, 0, 0,-1,-1,-1, 0, 1, 0, 0,-1, 0, 0, 1, 0, 0,-1,
 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,-1, 0, 1, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0,-1,-1,-1, 1,-1, 0, 0, 0, 0,-1, 0, 0, 0, 1,
 0,-1, 0, 0, 1, 0,-1, 0,-1, 0, 0, 0, 0, 1, 0,-1, 0,-1, 0, 0, 0, 0, 0, 0,-1,
 0, 0, 0, 0, 0, 0, 1,-1,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 1,-1,-1, 0,
 0, 0, 0, 0, 1,-1, 0,-1, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0,-1,-1, 0, 0,
 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 1,-1, 0,-1, 0, 0, 0,
 0, 0,-1, 0, 0, 0, 0, 1,-1, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0, 0, 0,-1, 1,
 0,-1,-1, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 1,-1, 0, 0, 1, 0, 0,-1, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0,-1, 0, 1,-1, 0,-1, 0, 0, 0,-1,-1, 0, 0, 0, 0, 0, 0,
-1, 1,-1, 0,-1, 0, 1, 0, 0,-1, 0, 0, 0, 0, 0, 0,-1, 0, 1, 0, 0,-1, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0,-1,-1, 0, 0,-1, 0, 0,-1, 0, 1,-1, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,-1, 0, 0, 1, 0, 0, 0,-1, 0, 0,-1, 0,-1, 1,-1,
 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0,-1, 0, 0, 1, 0, 0,
 0, 0,-1,-1, 1, 0, 0, 0, 0,-1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
-1, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0,-1, 0, 0, 0,-1, 0, 0,-1, 1,-1, 0, 0, 0, 1,
 0, 0, 0,-1, 0,-1,-1, 0, 0, 0, 0, 0, 0,-1,-1,-1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0,-1, 0, 1,-1, 0, 0, 0, 0, 1,-1, 0, 0, 0, 1, 0,-1, 0, 0,-1,-1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,-1, 0, 0,-1,
 0,-1, 0,-1,-1, 0, 0,-1, 1, 0, 0, 0, 0, 1,-1, 0, 0,-1, 0,-1, 0, 1,-1, 0, 0,
 0, 0,-1, 0, 0, 0, 0,-1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 1,
 0, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,-1, 1, 0,-1, 0,-1,-1, 0, 0, 0, 0,-1,
 0, 0, 0, 1, 0, 0, 0,-1, 1, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0,-1, 1, 0,-1, 0,-1,
 0, 0, 0, 0, 0, 0, 0, 1,-1, 0, 0,-1, 0, 0,-1, 0,-1, 1, 0, 0,-1, 0, 0, 1, 0,
 0, 0, 0, 0,-1, 0, 0, 0, 0,-1, 0, 0,-1, 1,-1, 0,-1, 0, 0, 0, 1, 0, 0,-1, 0,
 0, 0, 0, 0, 0, 0,-1, 0, 1, 0, 0, 0, 0, 0, 0, 0,-1, 0, 1,-1, 0,-1, 0, 1, 0,
 0, 0, 0,-1, 0, 0, 0, 0, 0,-1, 0, 0,-1, 0, 0, 1, 0, 0, 0, 0, 1, 0,-1,-1, 0,
 0, 0, 0, 0,-1, 0, 0,-1, 1, 0, 0,-1, 0, 1, 0,-1, 0, 0, 0, 0, 0, 0, 1, 0,-1,
 0, 0, 0,-1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,-1, 0, 1,-1,
-1, 0, 0, 0, 0, 0, 0,-1,-1, 0, 1, 0, 0,-1, 0, 0,-1, 1, 0, 0,-1, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,-1, 0, 0,-1, 1, 0, 0, 0, 0, 0, 0,-1,-1,
 1, 0, 0,-1, 0, 0,-1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0,-1,-1, 0, 1, 0,
-1, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 1, 0, 0, 0, 0, 0,-1,-1,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,-1, 0,-1, 0,-1,-1, 0, 0, 0, 0,
 1, 0, 0, 0, 0,-1,-1, 1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 1,-1, 0, 0,
-1, 0, 0, 0,-1, 0, 0, 1, 0, 0, 0, 0,-1, 0, 0, 1, 0, 0,-1,-1, 0,-1,-1, 0, 0,
 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,-1,-1, 0, 0, 0, 0, 0, 0, 0, 1,-1,-1, 0,-1, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0, 0,
 0, 1, 0, 0, 0,-1, 0, 0,-1,-1, 0, 0, 1,-1,-1, 0, 0, 0, 0, 1, 0,-1, 0, 0, 0,
 0, 0, 0, 0, 0,-1, 0, 1, 0,-1, 0,-1, 0, 0, 0, 0,-1, 0, 0,-1, 1, 0, 0, 0, 1,
 0,-1, 0, 0, 0, 0, 0,-1, 0, 1, 0, 0, 0, 0, 0, 0,-1, 1, 0,-1, 0,-1,-1, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0,-1,-1, 0,-1, 0,-1, 0, 0,-1, 0, 0, 1, 0, 0, 0,-1,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0, 0,-1,-1, 0,-1,-1, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
-1, 0,-1, 1, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0,-1,-1, 0, 0, 0,-1, 0, 1, 0, 0, 0,
 0, 0, 1,-1, 0, 0,-1, 0, 0,-1, 0, 0,-1, 0, 1, 0, 0,-1, 0, 0, 0, 0, 0, 1, 0,
-1, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0,-1, 0, 0, 0, 0,-1, 0, 0, 0,-1, 0, 1, 0,-1,
 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 1,-1, 0, 0,-1, 0, 0,-1, 0, 1, 0, 0,-1,
 0,-1, 0, 1, 0, 0, 0, 0, 0, 0,-1,-1, 0, 1, 0, 0, 0, 0,-1,-1, 1, 0, 0, 0, 0,
 0,-1, 0, 0,-1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 1, 0,-1, 0,
-1, 1, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0,-1, 1,-1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
 0,-1,-1, 0, 0,-1, 1, 0, 0, 0,-1,-1, 0, 0, 1, 0,-1, 0, 0, 0, 0, 0, 0, 0,-1,
 0,-1,-1, 0, 1, 0, 0,-1, 0, 0, 0, 0, 1, 0,-1,-1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 1, 0, 0, 0, 0,-1,-1, 1,-1,-1,-1, 0, 0, 0,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0, 1,-1, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0,-1, 0, 0, 0, 0,-1,-1,-1, 0, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0,-1, 0, 0,
 0,-1, 0, 0, 0, 0,-1,-1, 0, 0, 0, 1, 0, 0,-1, 0, 1, 0, 0, 0, 0,-1, 0, 0, 0,
-1, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 1,-1, 0, 0, 0,-1, 1, 0, 0,-1, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0,-1, 0, 0,-1, 1,-1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 1, 0,-1, 0,
-1, 0, 0,-1, 1, 0, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0,-1, 0, 0,-1, 1, 0,-1,-1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 0,-1,-1, 0, 1, 0, 0, 0,
 0, 0,-1,-1, 1, 0, 0, 0, 0, 0,-1, 0, 0, 1,-1, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0,
```

```
-1, 1, 0, 0,-1, 0,-1,-1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
-1, 1, 0, 0,-1, 0, 0, 0, 1, 0,-1,-1, 0,-1, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 1, 0,-1, 0, 0, 0, 1, 0, 0, 0, 0, 0,-1,-1,-1,-1,-1, 0,
 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1,-1,-1, 0,
-1, 0, 0, 1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0,-1,-1, 0,-1, 1, 0,-1, 0,-1,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,-1, 0, 0, 0, 0,-1, 0, 0, 0, 0,
 0, 0, 0, 0, 0,-1,-1, 1, 0,-1, 0,-1, 0, 1, 0, 0, 0, 0, 0,-1, 0,-1, 0, 0, 0,
-1,-1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,-1, 0, 0, 1, 0, 0,-1, 0, 0, 0, 0,
 0,-1, 0, 0, 0, 1, 0,-1,-1, 0, 0,-1, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0,-1, 0, 1,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,-1, 0,-1, 0, 0,-1, 0, 0, 1,-1,-1, 0,
 0, 0, 0, 0, 0,-1, 0, 0, 0, 1, 0,-1, 0, 0, 0, 0, 0, 0,-1, 0,-1, 0, 0, 0, 1,
 0, 0, 0,-1,-1,-1, 0, 1, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0,-1, 0, 1,
 0, 0,-1, 0, 0, 0, 0,-1, 0, 0,-1, 1,-1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 1, 0, 0,-1, 0, 0, 0, 0, 0, 0,-1, 0, 1,-1, 0, 0,-1,-1, 0,-1, 0, 0, 0,
-1,-1, 0, 0, 0, 0, 0,-1, 1, 0, 0, 0, 0,-1, 1,-1,-1, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0;
PESO = 10, 9.898876, 9.797752, 9.69663, 9.595506, 9.494382, 9.393258, 9.292135,
 9.191011, 9.089888, 8.988764, 8.88764, 8.786517, 8.685393, 8.58427, 8.483146,
 8.382023, 8.280899, 8.179775, 8.078651, 7.977528, 7.876405, 7.775281, 7.674157,
 7.573034, 7.47191, 7.370787, 7.269663, 7.16854, 7.067416, 6.966292, 6.865169,
 6.764045, 6.662921, 6.561798, 6.460674, 6.35955, 6.258427, 6.157303, 6.05618,
 5.955056, 5.853933, 5.752809, 5.651685, 5.550562, 5.449438, 5.348315, 5.247191,
 5.146068, 5.044944, 4.94382, 4.842697, 4.741573, 4.64045, 4.539326, 4.438202,
 4.337079, 4.235955, 4.134831, 4.033708, 3.932584, 3.831461, 3.730337, 3.629214,
 3.52809, 3.426966, 3.325843, 3.224719, 3.123596, 3.022472, 2.921348, 2.820225,
 2.719101, 2.617978, 2.516854, 2.41573, 2.314607, 2.213483, 2.11236, 2.011236,
 1.910113, 1.808989, 1.707865, 1.606742, 1.505618, 1.404495, 1.303371, 1.202247,
 1.101124, 1;
ENDDATA
MAX =  @SUM(DIA(J):@SUM (MILITAR(I): X(I,J)*PESO(I)*NOTA(I,J)));
@FOR (DIA(J):
@SUM( MILITAR(I): X(I,J))>=DEMANDA(J)
  );
 @FOR(MILITAR(I):
@SUM( DIA(J): X(I,J))<= CARGA(I);
@FOR (PARC(L):
@SUM( QUAN(M): X(I,L+M-1))<=1
);
X(I, 6)+X(I, 13)+X(I, 20)+X(I, 27)+X(I, 7)+X(I, 14)+X(I, 21)+X(I, 28)<= 2;
X(I, 6)+X(I, 13)+X(I, 20)+X(I, 27)<= 2;
X(I, 7)+X(I, 14)+X(I, 21)+X(I, 28)<= 2;
X(I, 6)+X(I, 13)+X(I, 7)+X(I, 14)<=1;
X(I, 13)+X(I, 20)+X(I, 14)+X(I, 21)<=1;
X(I, 20)+X(I, 27)+X(I, 21)+X(I, 28)<=1;
 );
X( 1, 1)+X( 1, 2)<=0;
X( 2, 1)+X( 2, 2)+X( 2, 3)<=0;
X( 4, 1)+X( 4, 2)<=0;
X( 6, 1)+X( 6, 2)+X( 6, 3)<=0;
X( 8, 1)<=0;
X( 9, 1)<=0;
X( 11, 1)+X( 11, 2)<=0;
X( 13, 1)<=0;
X( 14, 1)+X( 14, 2)<=0;
X( 15, 1)<=0;
X( 18, 1)<=0;
X( 20, 1)+X( 20, 2)<=0;
X( 22, 1)<=0;
X( 23, 1)+X( 23, 2)<=0;
X( 24, 1)+X( 24, 2)<=0;
X( 29, 1)+X( 29, 2)<=0;
X( 30, 1)+X( 30, 2)<=0;
X( 31, 1)+X( 31, 2)+X( 31, 3)<=0;
X( 33, 1)+X( 33, 2)+X( 33, 3)<=0;
X( 35, 1)+X( 35, 2)+X( 35, 3)<=0;
```

```
X( 38, 1)+X( 38, 2)<=0;
X( 39, 1)<=0;
X( 40, 1)+X( 40, 2)<=0;
X( 43, 1)+X( 43, 2)<=0;
X( 45, 1)+X( 45, 2)<=0;
X( 47, 1)+X( 47, 2)+X( 47, 3)<=0;
X( 49, 1)+X( 49, 2)<=0;
X( 51, 1)<=0;
X( 52, 1)+X( 52, 2)<=0;
X( 53, 1)<=0;
X( 55, 1)<=0;
X( 57, 1)+X( 57, 2)+X( 57, 3)<=0;
X( 58, 1)<=0;
X( 59, 1)<=0;
X( 60, 1)+X( 60, 2)<=0;
X( 62, 1)<=0;
X( 63, 1)<=0;
X( 67, 1)+X( 67, 2)<=0;
X( 74, 1)+X( 74, 2)<=0;
X( 75, 1)+X( 75, 2)<=0;
X( 76, 1)+X( 76, 2)<=0;
X( 77, 1)+X( 77, 2)<=0;
X( 81, 1)+X( 81, 2)<=0;
X( 82, 1)+X( 82, 2)<=0;
X( 84, 1)+X( 84, 2)+X( 84, 3)<=0;
X( 87, 1)+X( 87, 2)+X( 87, 3)<=0;
@FOR (DIA(J):
@FOR (MILITAR(I):
@BIN( X( I, J))
)
);
END
SET TERSEO 1
GO
DIVERT \TANIA\SOLU.TXT
NONZ X
RVRT
QUIT
```

Os quadros a seguir, indicam as respostas, depois de executado o programa de escala, considerando um período de 30 dias. Os hífens indicam os dias em que os 90 militares escolheram não trabalhar e os asteriscos indicam os dias em que escolheram trabalhar. Considerando o Militar 1, observa-se que foram atendidas suas expectativas tanto nos dias escolhidos para trabalhar quanto para os dias não escolhidos, lembrando que o número total de funcionários necessários por dia é de 19.

Quadro A1.1 – Resultados 24/24 Sem Finais de Semana Seguidos – Exemplo 1

4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	-	X*	-			X		X*					-			-	X	-	X*						X	
X*			X		X		-	-			X*			-		X		-	-	X*						
				-	-	X*						X	-		X*			-	-	X	-	X*			X	
X				-	X		X		-					-			X*					-	*		-	X
-		X		X*	-			X			-	X*				-	X		-	X*						-
X*		-				-				-		-			X*			X		X*			-	X		X
				-		X*			-	X	-			-		-				X			X*		X	
				-		-				-	-	X		X*		X			X*		-	X	-			X*
-	-			-						X		X		X*				-		-	X*			-	X	
	-		-			X				X*		-		X	-	*		-		X			-	X		X*
		X*		-	-	-	X		-	-						X					-	X*				X
		-	X*	-					*					X	-	X		-	X*		X			-	X	
-							-		-					X*		-	X		X*		X	-	-		X	
			-			X		X*	-			-	X		-	X*		X				X*		-	-	
						X*		-			X	-			X	-	X*		X			-	-	-		
		X*			-			X					-	X*		-	-	X	-	X	-	X*				
			X*				X	-	*	-	X			X*	-	X				-	X	-			-	
		X*			-						X*	-		-	X		-	X*		-			-	X		X
	X*	-	-			-		-	X					X*	-			X		X				-	X*	
			-								X		X*	-	-	-		X*	-	X		-	X			X*
	X*		X			-	X					X*		-	X		-	X*	-					-		
-	X*		X			-				-	X*					-	X		-	X					-	X*
	X*										X	-			X*		X	-	X	-	X*			-	-	-
X*							-	-	X*	-	X	-					-					X		X*		X
X*						-	X*		X	-	X		X	-			-		-				-	X*		
		-	-			-	X*		X	-	X				X*					-	-	X*				X
	-	-	-		X*			-			X*		X	-		-	X		X							X*
		-		X*					X*						X			-	X		-	-	-	X*	-	X
-				X*		-			X*		-	X	-					X*		-		-	X		X	
X			X			X*	-	-							X*		X	-			X*	-	-			
-	X	-						X*	-	X						-	-			X*			X		-	
			X*			-		X*	-		-					X	-					X*	-		-	
X*				-	X			-	X*		-	-	X						-					X*	-	
	-		X				X*						-		X*	-	X	-			X	-	-			
-	X*	-		-		X*			-	X						-		X*			-		X			
	X*		-	-			-		X	-		X*	-						X*				X	-		
-			X*				-	X		-		-	X*	-					X*	-			X			
-	X			-			X*		X			-	-	X*					-			-		X		
	X			X*	-					X*	-					-				-			-	X*	-	
			-		-	-			X			X	-	-	-			X*					X*			
-		X*	-			X		X*	-				X*		-			-	-				X			
	X			X*						X	-			-		-		-	-			-	X*			
X		-	X	-		X*	-					-				X	-			X*						
		-		X*			-		X					X*				-	X*		-	X	-	-		
X			X*				-	X*						X		-	X		-	X*		-		-		
X		X*	-			-			-		-	X*			-			X*			X			-		
X		-	*	-	X	-		X		X*			-								-		X*			
	-		X*	-	X	-		X*					-			X			-			-			X*	
X*		-	-	X					-	X		-	X*			-		X*		-						
X			-	-	X					X			X*								-		X*	-	-	

Continuação do Quadro A1.1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51	X					-	-		X*		X	-			-	X*			-		X*					X				-
52		X				-	X		-	X*					X		-	-	X*			-			-			X*		
53	X		X*		X		-		-	-		X*		-					X	-	X*						-			
54	X*		X		X		-	-						X*								X*		-		-	X	-	-	
55				X*		X			-	-	X*	-						X		X		-			X*	-			-	
56			-	X		*		X			-			X*			-	-		-	-	X				X		X*		
57	*	X		-	-			X		X			X*	-	-	X	-	X*										-		
58	X				X					X*	-		X		X*				-			-	-			X*	-	-		
59	X		X*		-	X								-		X*		-		-	X				-			-	X*	
60			X*		-			X			-		X*								-	X*		-		-	-	X		X
61						X*		-	-		-	X	-		X	-			X*		X		-	X*						X
62	X			-	X*		X				-	-		-	-	X*				X					X*			-	X	
63			-	X					X*			X		-		-	X*			X			X*		-			-	-	X
64			-		X*		X		X		X*	-			-			-		X	-		X*		X	-				
65		X*		-					X			X*	-		-			X		-	X			-		X*		-	X	
66					-			X			X*	-	X		-	X		-		*			-	X	-		X*		X	
67		X		-	-		*	X				-	-	X*			X		X	-			-				X*		X	
68					-					X*		-	X	-	X*			-				X			X	-	X*	-	X	
69	X*				X		X			-	-			-	X*		X		-	-			X*		-				X	
70			-	X	-	-		X*		X	-		X			X*		-	-			X*						X		
71								-		X*		X		X	-	-	X*	-	-	-		X			X*			X		
72			X*			-			X*	-	X			X*			X		X			-		X			-	-	-	
73		X				X*		-	X		-				-			X		-	-				X*		X	-		X*
74		X			-			X	-	X			X*	-			X			*	-				-	X*		X	-	
75		X			X		*		X	-			-	X*	-	X				-					X*		-	X	-	
76		-	X*			-	X				X*				-	X		-	X*		-	-		X			X			
77		X				X*			-	X		-	-		X*			X		X	-	-	X*						-	
78		X*	-			X	-	X			X			-	X*			-		-	-			X*				X		
79						X		X	-	X*			-				X*		-	-	X	-	X		X*	-				
80			X*		X		X				X*		-				X*			X			-	-	-	-	-		X	
81			X*	-					X	-		X		X					X*	-	-	-	X	-			X*			
82			-	X				X		-	-		-	X*		-	X	-	X*							X			X*	
83			X*	-		X		X	-								X		-	-	X*		-	X	-		*		X	
84			-	X	-	X		X	-	-		X*						X*		X			-			X*			-	
85					-				X*		-	-		X	-	X*					-	X			X	-		X*		X
86			X*			X			X		X*	-		-			-			X*	-	-				X			-	X
87			X*		-	X			X		X	-		-				X*		X		-	-	-		X*				
88		X		-	X						-		X*			-			X		-			-	X*	-		X		X*
89						X*			X		X*			-		X				X	-		X*	-		X	-	-		-
90				-	-		X				-	X*			X		-	X*	-	-	X		X*		X					

Quadro A1.2 – Resultados 24/24 Com Finais de Semana Seguidos – Exemplo 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			-		-	X*	-					X*			X		-			-		-	X*				X			X
2			-	X*			X				-	-			X*			-			X	-	-	X*					X	
3				X				-	-	X*						X	-		X*			-	-		-	X*			X	
4		X*	-					-			X		-			X		-			X*					-	X*		-	X
5				-				X*	-			X			-	X*				-	X		-	X*					X	-
6			-	X*		-	X			-	X			-	X	-			X*					X*			-			
7		X*	-					-		X*			-		-	X		-	X	-			X				X*			
8	X						X	-		-			X	-	-			X*					X*		-		-			X*
9	X*			-	-	X		-	X			X						X*				-		-	X*			-		
10					-		-	X						X*		-	X		-	X*		-	X				-			X*
11	*	X				X*		-	-	-	X		-	-	X					X					-	X*				
12			-			-	X*	-					X*			X			-			-	X*			X		-	X	
13			X*	-					X		-		-	X				X*		-			X*			-	-	X		
14						X	-					X*	-			-	X		-	X*						X*		-	-	X
15			X*							X*		-	X			-	X			-	X*				X	-	-	-		
16						X*			-	X			X				-	X*		-	-		-		-	X*			X	
17				X			X*					-	X*	-				X*	-			X		-	X	-			-	
18						X*			-			X			X*	-		-			-	X*		-	X		-	X		
19					X*	-	-			-	X	-		X				X*	-							X		-	X*	
20					X		-	X				X					X*	-	-	-		X*	-			-				X*
21			-		X*					-			X			X*		-			-	X*	-			X		-	X	
22				-	X*					-				-	X*					-	X		-	X			X		-	X*
23		X			X*				X						X	-			X*			-		-	X*			-	-	-
24			-	X*					X		-	-	X*	-		-			X		-	X						X*		
25				X*				X		-	X*			-				-		X	-		-		X		-	X*		
26				X		-	-			-	X*			-	X				X*					-	-	X*			X	
27				X	-	-	-		X*			-			X*			-		-	X				X					X*
28	X			X		-		X*					X*						X			-			-	-	-	X*	-	
29				-	X			X*		-			X*		-		-					X*		-	X	-		X		
30			-				X			X*	-	-			X				X*			-			X*	-	-	X		
31			X*	-		-			X			X*	-				X			-	-			X*			X		-	
32	X			X			X*			-		X*	-		-	X					-					X*	-		-	
33				X*				-	X			-	X*		-	-	X						-	X				X*	-	
34		X*			-		X				X*						-		X*	-		-	X			-	-	X		
35				-	X*	-		-		X*			-	X			X			-		X*			-		X			
36					X*		-	-	X		-		X	-		X*	-			X			X*					-		
37	X			-			X*			X	-			-		-	X*	-					X*	-				X		
38			X*	-			X	-			X*			X		-	-	X*					-		X	-				
39					X			X*	-					X*	-		X			-				-	X		-	X*	-	
40			X*				-	X	-	-	X			X			-	-	-			X*					X*			
41				-		X*	-					X*	-				X*		-	X		-	-	X					X	
42			X*					X*				X			-		X	-		-	X	-	-			-	X*			
43		X*	-		X	-		-		X*	-					-				X	-			X*				X		
44					X	-		X*			-	X						X*				-	X*		-		-	-	X	
45		X	-				X*				-	X*				X				-	X		-	X*		-		-		
46			X			X*	-			-			-		-	X*			-			X*					X	-		X
47			-	X		-	*	-	X	-				X*			-	X				X			-		X*			
48					-		X*	-		-		X*					-	X			X		-	X		-			X*	
49				X*		-	-						-	X		-	X*			-		X*		-			X			X
50	X*		-	X			-	-				X					X*			X					-		X*	-	-	

Continuação do Quadro A1.2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51	X				X	-	-		X*			-			-	X*			-		X*			X						-
52		X				-	X		-	X*							-	-	X*			-		X	-			X*		
53			X*				-	X	-	-		X*		-				X		-	X*						-	X		
54	X*				X		-	-						X*					X			X*		-	X	-		-	-	
55				X*			X		-	-	X*	-					X			X		-			X*	-			-	
56	X		-	X		*		X			-			X*			-	-		-	-			X				X*		
57	*		X	-	-		X						X*	-	-		-	X*				X				X		-		
58	X									X*	-				X*			X	-		X	-	-			X*	-	-		
59			X*		-	X			X					-		X*		-		-	X				-			-	X*	
60			X*		-					X	-		X*			X					-	X*		-		-	-	X		
61			X			X*		-	-	X	-		-		X	-			X*				-	X*			X			
62	X			-	X*				X		-	-	X	-	-	X*				X					X*			-		
63	X		-	X					X*				X	-		-	X*			X			X*		-			-	-	
64	X		-		X*			X			X*	-			-			-		X	-		X*			-		X		
65		X*		-		X			X			X*	-		-	X				-	X			-		X*		-		
66				X	-						X*	-			-	X		-		X*			-	X	-		X*			X
67		X		-	-		*	X				-	-	X*				X		-			-	X			X*			X
68	X				-	X				X*		-		-	X*			-	X							-	X*	-		X
69	X*					X			X	-	-			-	X*			X	-	-			X*		-				X	
70	X		-	X	-	-		X*			-	X				X*		-	-			X*						X		
71	X				X			-		X*			X		-	-	X*	-	-	-	X				X*					
72			X*			-			X*	-				X*					X			-	X			X	-	-	-	X
73	X					X*		-	X		-		X		-					-	-	X			X*			-		X*
74		X			-				-	X			X*	-		X				X*	-				-	X*			-	X
75		X					X*			-	X		-	X*	-		X			-		X			X*		-		-	
76		-	X*			-	X				X*			X	-			-	X*		-	-	X						X	
77		X				X*			-	X		-	-		X*					X	-	-	X*			X			-	
78		X*	-			X	-			X				-	X*			-	X	-	-			X*			X			
79	X				X				-	X*			-	X			X*		-	-	X	-			X*	-				
80			X*					X			X*		-	X			X*			X			-	-	-	-	-		X	
81			X*	-				X		-	X			X					X*	-	-	-	X	-			X*			
82		X	-		X			X		-	-		-	X*		-		-	X*			X							X*	
83			X*	-					-	X					X			X	-	-	X*		-	X	-		X*			
84			-		-		X		-	-		X*			X			X*			X		-			X*			-	X
85		X			-	X			X*		-	-			-	X*			X		-	X				-		X*		
86			X*					X			X*	-		-	X		-			X*	-	-		X					-	X
87			X*		-	X					X	-		-				X*			X	-	-	-		X*			X	
88		X		-	X				X		-		X*			-				X	-			-	X*	-				X*
89		X				X*					X*			-						X	-		X*	-		X	-	-	X	-
90		X		-	-		X				-	X*					-	X*	-	-			X*			X			X	

Quadro A1.3 – Resultados 24/48 Sem Finais de Semana Seguidos – Exemplo 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		X	-		-	X*	-					X*				X	-			-		-	X*						X	
2			-	X*			X				-	-			X*			-		X		-	-	X*					X	
3				X				-	-	X*				X			-		X*			-	-		-	X*			X	
4		X*	-					-			X		-	X				-	X		*	X				-	X*		-	
5				-	X			X*	-			X			-	X*				-	X		-	X*						-
6			-	X*		-		X		-			X	-		-			X*					X*			-		X	
7		X*	-					-		X*			-	X	-		X	-		-							X*			X
8					X			-	X	-			X	-	-			X*					X*		-		-			X*
9	X*			-	-	X		-	X									X*			X	-		-	X*			-		
10			X		-		-		X					*		-			-	X*		-	X			X	-			X*
11	*	X				X*		-	-	-	X		-	-	X					X					-	X*				
12			-	X		-	X*	-					*			X			-	X		-	X*					-	X	
13			X*	-					X		-		-	X				X*		-			X*			-	-	X		
14		X				X	-					X*	-		X	-			-	X*						X*		-	-	
15			X*				X			X*		-				-	X			-	X*					-	-	-	X	
16						X*			-			X			X		-	X*		-	-		-		-	X*			X	
17							X*			X		-	*	-	X			X*	-		X			-	X	-			-	
18						X*			-			X			X*	-		-	X		-	X*		-			-	X		
19					X*	-	-	X		-		-						X*	-		X					X		-	X*	
20							-				X			X			X*	-	-	-		X*	-			-	X			X*
21			-		X*			X		-			X			X*		-			-	X*	-				X	-		
22				-	X*					-	X			-	X*					-	X		-	X					-	X*
23		X			X*			X					X			-			X*			-		-	X*			-	-	-
24			-	X*					X		-	-	X*	-		-					-	X			X			X*		
25				X*						-	X*			-	X			-	X		-		-		X		-	X*		
26	X			X		-	-			-	X*			-	X				X*					-	-	X*				
27				X	-	-	-		X*			-			X*			-		-	X				X					X*
28	X					-		X*					X*						X			-	X		-	-	-	X*	-	
29				-	X			X*		-			X*		-	X	-					X*		-		-		X		
30			-	X						X*	-	-	X						X*			-			X*	-	-	X		
31			X*	-		-	X					X*	-				X			-	-			X*				X	-	
32							X*			-		X*	-		-		X			X	-					X*	-		-	X
33				X*				-		X		-	X*		-	-	X						-		X			X*	-	
34		X*			-			X			X*			X			-		X*	-		-				-	-	X		
35				-	X*	-		-		X*			-	X			X			-		X*			-		X			
36					X*		-	-			-	X		-		X*	-			X			X*					-	X	
37	X			-			X*				-	X		-		-	X*	-		X			X*	-						
38			X*	-			X	-			X*					-	-	X*			X		-	X		-				
39	X				X			X*	-		X			X*	-					-				-			-	X*	-	
40			X*				-	X	-	-	X			X			-	-	-			X*					X*			
41	X			-		X*	-					X*	-				X*		-	X		-	-	X						
42			X*					X*					X		-	X		-		-		-	-	X		-	X*			
43		X*	-			-	X	-		X*	-					-	X			X	-			X*						
44					X	-		X*			-			X				X*				-	X*		-	X	-	-		
45		X	-				X*				-	X*			X					-	X		-	X*		-		-		
46	X					X*	-			-	X		-		-	X*			-			X*					X	-		
47			-	X		-	*	-		-	X			X*			-	X				X			-		X*			
48	X			X	-		X*	-		-		X*					-			X			-			-			X*	
49				X*		-	-	X					-	X		-	X*			-		X*		-				X		
50	X*		-	X			-	-	X				X				X*								-		X*	-	-	

Continuação do Quadro A1.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51					X	-	-		X*			-			-	X*			-		X*					X			X	-
52		X				-			-	X*			X			X	-	-	X*			-			-			X*		
53			X*				-		-	-		X*		-				X		-	X*			X			-		X	
54	X*				X		-	-						X*			X					X*		-		-	X	-	-	
55	X			X*			X		-	-	X*	-									X	-			X*	-			-	
56			-	X		*		X			-			X*			-	-	X	-	-			X				X*		
57	*		X	-	-			X					X*	-	-		-	X*						X			X	-		
58	X						X			X*	-				X*				-	X		-	-			X*	-	-		
59			X*		-					X			X	-		X*		-		-				X	-			-	X*	
60			X*		-					X	-		X*					X			-	X*		-		-	-	X		
61			X			X*		-	-		-	X	-			-			X*				-	X*			X			X
62	X			-	X*					X	-	-		-	-	X*					X				X*			-		X
63	X		-			X			X*					-		-	X*			X			X*		-	X		-	-	
64		X	-		X*			X			X*	-			-			-		X	-		X*			-				X
65		X*		-		X						X*	-		-			X		-	X			-		X*		-	X	
66	X			X	-		X				X*	-			-	X		-		X*			-	X	-		*			
67		X		-	-		*		X			-	-	X*				X		-		X	-				X*			X
68					-	X				X*		-		-	X*			-	X					X		-	X*	-		X
69	X*					X			X	-	-			-	X*				-	-			X*		-		X			X
70			-	X	-	-		X*			-		X			X*		-	-			X*			X			X		
71	X				X			-		X*					-	-	X*	-	-	-	X				X*				X	
72			X*			-			X*	-				X*				X				-	X			X	-	-	-	X
73		X				X*		-	X		-	X			-		X			-	-				X*			-		X*
74		X			-		X		-	X			*	-		X				X*	-		X		-	X*			-	
75		X					*		X	-			-	X*	-				X	-		X			X*		-	X	-	
76		-	X*			-	X				X*				-	X		-	X*		-	-			X					X
77		X				X*			-			-	-		X*					X	-	-	X*			X			-	X
78		X*	-			X	-		X			X		-	X*			-		-	-			X*			X			
79	X				X				-	X*			-				X*		-	-	X	-			X*	-			X	
80			X*					X			X*		-	X			X*					X	-	-	-	-	-	X		
81			X*	-					X	-			X			X			X*	-	-	-	X	-			X*			
82			-		X			X		-	-		-	X*		-		-	X*			X			X				X*	
83			X*	-			X		-	X					X			X	-	-	X*		-	X	-		*			
84			-	X	-		X		-	-		X*			X			X*			X		-			X*			-	
85					-				X*		-	-	X		-	X*			X		-	X			X	-		X*		
86			X*				X				X*	-		-	X		-			X*	-	-	X						-	X
87			X*		-	X				X		-		-				X*			X	-	-	-		X*			X	
88		X		-	X				X		-		X*			-	X				-			-	X*	-				X*
89						X*					X*			-		X				X	-		X*	-		X	-	-	X	-
90	X			-	-	X			X		-	X*					-	X*	-	-			X*				X			

Quadro A1.4 – Resultados 24/48 Com Finais de Semana Seguidos – Exemplo 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			-		-	X*	-					X*			X		-			-		-	X*				X			X
2			-	X*			X				-	-			X*			-			X	-	-	X*					X	
3				X				-	-	X*						X	-		X*			-	-		-	X*			X	
4		X*	-					-			X		-			X		-			X*					-	X*		-	X
5				-				X*	-			X			-	X*				-	X		-	X*					X	-
6			-	X*		-	X			-	X			-	X	-			X*					X*			-			
7		X*	-					-		X*			-		-	X		-	X	-			X				X*			
8	X						X	-		-			X	-	-			X*					X*		-		-			X*
9	X*			-	-	X		-	X			X						X*				-		-	X*			-		
10					-		-	X						X*		-	X		-	X*		-	X				-			X*
11	*	X				X*		-	-	-	X		-	-	X					X					-	X*				
12			-			-	X*	-					X*			X			-			-	X*			X		-	X	
13			X*	-					X		-		-	X				X*		-			X*			-	-	X		
14						X	-					X*	-			-	X		-	X*						X*		-	-	X
15			X*							X*		-	X			-	X			-	X*				X	-	-	-		
16						X*			-	X			X				-	X*		-	-		-		-	X*			X	
17				X			X*					-	X*	-				X*	-			X		-	X	-			-	
18						X*			-			X			X*	-		-			-	X*		-	X		-	X		
19					X*	-	-			-	X	-		X				X*	-							X		-	X*	
20					X		-	X				X					X*	-	-	-		X*	-			-				X*
21			-		X*					-			X			X*		-			-	X*	-			X		-	X	
22				-	X*					-				-	X*					-	X		-	X			X		-	X*
23		X			X*				X						X	-			X*			-		-	X*			-	-	-
24			-	X*					X		-	-	X*	-		-			X		-	X						X*		
25				X*				X		-	X*			-				-		X	-		-		X		-	X*		
26				X		-	-			-	X*			-	X				X*					-	-	X*			X	
27				X	-	-	-		X*			-			X*			-		-	X				X					X*
28	X			X		-		X*					X*						X			-			-	-	-	X*	-	
29				-	X			X*		-			X*		-		-					X*		-	X	-		X		
30			-				X			X*	-	-			X				X*			-			X*	-	-	X		
31			X*	-		-			X			X*	-				X			-	-			X*			X		-	
32	X			X			X*			-		X*	-		-	X					-					X*	-		-	
33				X*				-	X			-	X*		-	-	X						-	X				X*	-	
34		X*			-		X				X*						-		X*	-		-	X			-	-	X		
35				-	X*	-		-		X*			-	X			X			-		X*			-		X			
36					X*		-	-	X		-		X	-		X*	-			X			X*					-		
37	X			-			X*			X	-			-		-	X*	-					X*	-				X		
38			X*	-			X	-			X*			X		-	-	X*					-		X	-				
39					X			X*	-					X*	-		X			-				-	X		-	X*	-	
40			X*				-	X	-	-	X			X			-	-	-			X*					X*			
41				-		X*	-					X*	-				X*		-	X		-	-	X					X	
42			X*					X*				X			-		X	-		-	X	-	-			-	X*			
43		X*	-		X	-		-		X*	-					-				X	-			X*				X		
44					X	-		X*			-	X						X*				-	X*		-		-	-	X	
45		X	-				X*				-	X*				X				-	X		-	X*		-		-		
46			X			X*	-			-			-		-	X*			-			X*					X	-		X
47			-	X		-	*	-	X	-				X*			-	X				X			-		X*			
48					-		X*	-		-		X*					-	X			X		-	X		-			X*	
49				X*		-	-						-	X		-	X*			-		X*		-			X			X
50	X*		-	X			-	-				X					X*			X					-		X*	-	-	

Continuação do Quadro A1.4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51	X				X	-	-		X*			-			-	X*			-		X*			X						-
52		X				-	X		-	X*							-	-	X*			-		X	-			X*		
53			X*				-	X	-	-		X*		-				X		-	X*						-	X		
54	X*				X		-	-						X*					X			X*		-	X	-		-	-	
55				X*			X		-	-	X*	-					X			X		-			X*	-			-	
56	X		-	X		*		X			-			X*			-	-		-	-			X				X*		
57	*		X	-	-		X						X*	-	-		-	X*				X				X		-		
58	X									X*	-				X*			X	-		X	-	-			X*	-	-		
59			X*		-	X			X					-		X*		-		-	X				-			-	X*	
60			X*		-					X	-		X*			X					-	X*		-		-	-	X		
61			X			X*		-	-	X	-		-		X	-			X*				-	X*			X			
62	X			-	X*				X		-	-	X	-	-	X*				X					X*			-		
63	X		-	X					X*				X	-		-	X*			X			X*		-			-	-	
64	X		-		X*			X			X*	-			-			-		X	-		X*			-		X		
65		X*		-		X			X			X*	-		-	X				-	X			-		X*		-		
66				X	-						X*	-			-	X		-		X*			-	X	-		X*			X
67		X		-	-		*	X				-	-	X*				X		-			-	X			X*			X
68	X				-	X				X*		-		-	X*			-	X							-	X*	-		X
69	X*					X			X	-	-			-	X*			X	-	-			X*		-				X	
70	X		-	X	-	-		X*			-	X				X*		-	-			X*						X		
71	X				X			-		X*			X		-	-	X*	-	-	-	X				X*					
72			X*			-			X*	-				X*					X			-	X			X	-	-	-	X
73	X					X*		-	X		-		X		-					-	-	X			X*			-		X*
74		X			-				-	X			X*	-		X				X*	-				-	X*			-	X
75		X					X*			-	X		-	X*	-		X			-		X			X*		-		-	
76		-	X*			-	X				X*			X	-			-	X*		-	-	X						X	
77		X				X*			-	X		-	-		X*					X	-	-	X*			X			-	
78		X*	-			X	-			X				-	X*			-	X	-	-			X*			X			
79	X				X				-	X*			-	X			X*		-	-	X	-			X*	-				
80			X*					X			X*		-	X			X*			X			-	-	-	-	-		X	
81			X*	-				X		-	X			X					X*	-	-	-	X	-			X*			
82		X	-		X			X		-	-		-	X*		-		-	X*			X							X*	
83			X*	-					-	X					X			X	-	-	X*		-	X	-		X*			
84			-		-		X		-	-		X*			X			X*			X		-			X*			-	X
85		X			-	X			X*		-	-			-	X*			X		-	X				-		X*		
86			X*					X			X*	-		-	X		-			X*	-	-		X					-	X
87			X*		-	X					X	-		-				X*			X	-	-	-		X*			X	
88		X		-	X				X		-		X*			-				X	-			-	X*	-				X*
89		X				X*					X*			-						X	-		X*	-		X	-	-	X	-
90		X		-	-		X				-	X*					-	X*	-	-			X*			X			X	

Quadro A1.5 – Resultados 24/72 Sem Finais de Semana Seguidos – Exemplo 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			-		-	X*	-				X	*			X		-		X	-		-	X*				X			
2			-	X*					X		-	-			X*			-		X		-	-	X*				X		
3		X				X		-	-	X*				X			-		X*			-	-		-	X*				
4		*	-	X				X-				X	-				X	-			X*					-	X*		-	
5			X	-				X*	-			X			-	X*				-			-	X*				X		-
6			-	X*		-		X		-			X	-		-			X*					X*			-	X		
7		X*	-					-		X*			-	X	-			-	X	-			X				X*			
8		X					X	-		-			X	-	-			X*					X*		-		-			X*
9	*	X		-	-	X		-		X				X				X*				-		-	X*			-		
10				X	-		-		X					X*		-			-	X*		-				X	-			X*
11	*					X*		-	-	-	X		-	-	X					X					-	X*				X
12		X	-			-	X*	-					X*				X		-			-	X*					-	X	
13			X*	-					X		-		-	X				X*		-			X*			-	-	X		
14			X				-	X				X*	-			-			-	X*						X*		-	-	X
15			X*							X*		-		X		-				-	X*				X	-	-	-	X	
16		X				X*			-					X			-	X*		-	-	X	-		-	X*				
17		X					X*					-	X*	-				X*	-				X	-		-			-	X
18		X				X*			-	X					X*	-		-			-	X*		-			-	X		
19	X				X*	-	-			-		-	X					X*	-					X				-	X*	
20				X			-		X				X				X*	-	-	-		X*	-			-				X*
21	X		-		X*				X	-						X*		-			-	X*	-			X		-		
22				-	X*				X	-				-	X*				X	-			-		X				-	X*
23					X*				X				X			-	X		*		X	-		-	X*			-	-	-
24			-	X*				X			-	-	X*	-		-	X				-		X					X*		
25				X*						-	X*			-		X		-		X	-		-	X			-	X*		
26	X				X	-	-			-	X*			-	X				X*					-	-	X*				
27				X	-	-	-		X*			-			X*			-		-	X				X					X*
28				X		-		X*					X*						X			-		X	-	-	-	X*	-	
29			X	-				X*		-			X*		-		-	X				X*		-		-	X			
30			-			X				X*	-	-		X					X*			-			X*	-	-		X	
31			*	-		-		X				X*	-			X				X-	-			X*				X	-	
32			X				X*			-		X*	-		-	X				X	-					X*	-		-	
33				X*				-	X			-	X*		-	-	X						-	X				X*	-	
34		X*			-		X				X*				X		-		X*	-		-		X		-	-			
35				-	X*	-		-		X*			-	X				X		-		X*			-		X			
36	X				X*		-	-			-	X		-		X*	-						X*				X	-		
37		X		-			X*				-	X		-		-	X*	-					X*	-			X			
38			X*	-			X	-			X*					-	-	X*				X	-			-	X			
39		X						X*	-					X*	-			X		-		X		-			-	X*	-	
40			X*				-	X	-	-		X				X	-	-	-			X*					X*			
41		X		-		X*	-					X*	-				X*		-		X	-	-		X					
42			X*					X*				X			-		X	-		-	X	-	-			-	X*			
43		*	-		X	-		-		X*	-				X	-				X	-			X*				X		
44	X					-		X*			-		X					X*				-	X*		-		-	-	X	
45			-				X*				-	X*				X				X-			-	X*		-		-	X	
46		X				X*	-			-	X		-		-	X*			-			X*				X		-		
47			-	X		-	*	-	X	-				X*			-		X				X		-		X*			
48					-		X*	-		-		X*				X	-			X			-	X		-			X*	
49				X*		-	-	X				X	-			-	X*			-		X*		-		X				
50	X*		-		X		-	-			X						X*				X				-		X*	-	-	

Continuação do Quadro A1.5

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51					X	-	-		X*			-			-	X*			-		X*				X				X	-
52					X	-			-	X*				X			-	-	X*			-	X		-			X*		
53			X*				-	X	-	-		X*		-		X				-	X*				X		-			
54	X*						-	-	X					X*				X				X*		-		-	X	-	-	
55				X*					-	-	X*	-			X					X		-			X*	-			-	X
56	X		-			X*				X	-			X*			-	-		-	-	X				X		*		
57	*			-	-		X						X*	-	-		-	X*				X				X		-		X
58		X				X				X*	-				X*				-	X		-	-			X*	-	-		
59			X*		-		X				X			-		X*		-		-	X				-			-	X*	
60			X*		-				X		-		X*					X			-	X*		-		-	-	X		
61	X					X*		-	-	X	-		-		X	-			X*				-	X*				X		
62				-	X*				X		-	-	X	-	-	*	X				X				X*			-	X	
63			-	X					X*				X	-		-	X*				X		*		-	X		-	-	X
64	X		-		X*						X*	-			-	X		-		X	-		*	X		-		X		
65		X*		-			X					X*	-		-		X			-	X			-		X*		-		X
66			X		-		X				X*	-			-	X		-		X*			-	X	-		*		X	
67			X	-	-		X*					-	-	X*				X		-		X	-			X	*			X
68		X			-	X				X*		-		-	X*			-	X				X			-	X*	-		
69	X*				X				X	-	-		X	-	*			X	-	-			X*		-		X			
70			-	X	-	-		X*			-	X				X*		-	-			X*				X				X
71		X				X		-		X*					-	-	X*	-	-	-	X				X*				X	
72	X		*		X	-			X*	-				X*						X		-		X			-	-	-	X
73	X					X*		-		X	-			X	-				X	-	-				X*			-		X*
74				X	-			X	-			X	*	-		X				X*	-				-	X*			-	X
75			X				X*			-	X		-	*	-	X				-	X				X*		-		-	X
76		-	X*			-	X				X*				X-			-	X*		-	-	X				X			
77			X			*	X		-		X	-	-		X*				X		-	-	X*				X		-	
78		X*	-			X	-			X				-	X*			-	X	-	-			X*				X		
79		X				X			-	X*			-				X*		-	-	X	-			X*	-			X	
80			X*				X				X*		-				X*				X		-	-	X-	-	-		X	
81			X*	-			X			-	X				X				X*	-	-	-	X	-			X*			
82			-	X				X		-	-	X	-	*		-	X	-	*		X				X				X*	
83	X		*	-	X				-	X				X				X	-	-	*	X	-		-		X*			
84			-	X	-			X	-	-		X*						X*				X	-			X*			-	X
85	X				X-				X*		-	-			-	X*				X	-			X		-		X*		
86	X		*		X						X*	-		-	X		-			X*	-	-		X				X	-	
87			*	X	-			X				-	X	-			X	*			X	-	-	-	X	*			X	
88	X			-	X				X		-		X*			-				X	-			-	X*	-			X	*
89	X					X*					X*			-	X				X		-		X*	-			-	-	X	-
90	X			-	-	X				X	-	*		X			-	X*	-	-			X*						X	

Quadro A1.6 – Resultados 24/72 Com Finais de Semana Seguidos – Exemplo 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			-		-	X*	-					X*				X	-			-	X	-	*		X				X	
2			-	X*				X			-	-			X*			-		X		-	-	X*					X	
3	X				X			-	-	X*				X			-		X*			-	-		-	X*				
4		*	-				X	-				X	-			X		-			X*				X	-	*		-	X
5		X		-				X*	-			X			-	X*				-			-	X*				X		-
6			-	X*		-		X		-			X	-		-			X*					X*			-	X		
7		X*	-					-		X*			-	X	-			-	X	-			X				X*			
8				X				-	X	-			X	-	-			X*					X*		-		-			X*
9	*	X		-	-			-	X					X				X*				-		-	X*			-	X	
10		X			-	X	-					X		*		-			-	X*		-		X			-			X*
11	*					X*		-	-	-		X	-	-		X				X					-	X*				X
12	X		-		X	-	*	-	X				X*						-			-	X*				X	-		
13			X*	-				X			-		-	X				X*		-			X*			-	-	X		
14				X			-	X				X*	-			-			-	X*						X*		-	-	X
15			X*							X*		-			X	-				-	X*				X	-	-	-		X
16		X				X*			-		X						-	X*		-	-	X	-		-	X*				
17				X			*	X				-	X*	-				X*	-				X	-		-	X		-	
18		X				X*			-		X				X*	-		-				X*		-			-	X		
19					X*	-	-		X	-		-		X				X*	-				X					-	X*	
20					X		-		X				X				X*	-	-	-		X*	-			-				X*
21	X		-		X*					-		X				X*		-			-	X*	-				X	-		
22				-	X*					-	X			-	X*					-	X		-			X			-	X*
23					X*				X				X			-			X*			-	X	-	*		X	-	-	-
24			-	X*				X			-	-	X*	-		-	X				-	X						X*		
25				X*						-	X*			-	X			-	X		-		-	X			-	X*		
26	X				X	-	-			-	X*			-	X				X*					-	-	X*				
27	X				-	-	-		X*			-			X*			-		-	X					X				X*
28				X		-		X*					X*				X					-	X		-	-	-	X*	-	
29			X	-				X*		-			X*		-		-	X				X*		-		-	X			
30			-		X					X*	-	-		X					X*			-			X*	-	-		X	
31			*	-	X	-				X		*	-	X					X	-	-			X*				X	-	
32	X						X*			-		X*	-		-	X				X	-					X*	-		-	
33				X*				-	X			-	X*		-	-	X					X	-					X*	-	
34		X*			-	X					X*				X		-		X*	-		-	X			-	-			
35				-	X*	-		-		X*			-	X				X		-		X*			-		X			
36	X				X*		-	-		X	-			-		X*	-						X*				X	-		
37	X			-			X*				-	X		-		-	X*	-					X*	-			X			
38			X*	-			X	-			X*				X	-	-	*		X			-	X		-				
39				X				X*	-					X*	-			X		-		X		-			-	X*	-	
40			X*				-	X	-	-				X			-	X-	-			X*					X*			
41	X			-		X*	-					X*	-				X*		-		X	-	-		X					
42			X*					X*						X	-			-	X	-		-	-	X		-	*	X		
43		*	-		X	-		-		X*	-			X		-		X			-			X*				X		
44	X					-		X*			-		X					X*				-	X*		-		-	-	X	
45			-				X*				-	X*				X				X-			-	X*		-		-	X	
46		X				X*	-			-	X		-		-	X*			-		X	*			X			-		
47			-	X		-	*	-	X	-				X*			-		X				X		-		X*			
48					-		X*	-		-		X*				X	-			X			-	X		-			X*	
49				X*		-	-	X					X-			-	X*			-		X*		-			X			
50	X*		-				-	-	X				X				X*					X			-		X*	-	-	

Continuação do Quadro A1.6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
51					X	-	-		X*			-			-	X*			-		X*				X				X	-
52				X		-			-	X*				X			-	-	X*			-		X	-			X*		
53			X*				-	X	-	-		X*		-			X			-	X*					X	-			
54	X*						-	-		X				X*				X				X*		-		-	X	-	-	
55		X		*			X		-	-	X*	-			X					X		-			X*	-			-	
56			-	X		*			X		-			X*			-	-	X	-	-			X				X*		
57	*			X-	-			X					X*	-	-		-	X*				X				X		-		
58		X				X				X*	-				X*				-	X		-	-			X*	-	-		
59			X*		-		X					X		-		X*		-		-	X				-	X		-	*	
60			X*		-				X		-		X*				X				-	X*		-		-	-	X		
61		X				X*		-	-	X	-		-		X	-			X*				-	X*				X		
62		X		-	*	X				X	-	-		-	-	X*				X					X*			-	X	
63			-	X					X*				X	-		-	X*					X	*		-	X		-	-	X
64	X		-		X*				X		*	-		X	-			-	X		-		X*			-		X		
65		X*		-			X					X*	-		-		X			-	X			-		X*		-		X
66		X			-	X					X*	-			-	X		-		X*			-	X	-		*		X	
67			X	-	-		X*				X	-	-	*			X			-	X		-		X		*		X	
68	X				-	X				X*		-		-	X*			-	X				X			-	X*	-		
69	X*				X				X	-	-		X	-	*			X	-	-			X*		-		X			
70			-	X	-	-		X*			-	X				X*		-	-	X		*		X					X	
71	X					X		-		X*					-	-	X*	-	-	-	X				X*				X	
72	X		*		X	-			X*	-				X*				X				X-				X	-	-	-	
73		X				X*		-		X	-				-	X				X-	-				X*			-		X*
74			X		-		X		-			X	*	-		X				X*	-				-	X*			-	X
75			X				X*			-	X		-	*	-		X			-	X				X*		-		-	X
76		-	X*			-	X				X*				-	X		-	*	X	-	-		X					X	
77			X			*	X		-		X	-	-		X*				X		-	-	X*				X		-	
78		X*	-			X	-			X				-	X*			-	X	-	-			X*				X		
79		X				X			-	X*			-				X*		-	-	X	-			X*	-			X	
80			X*				X				X*		-				X*				X		-	-	X-	-	-		X	
81			X*	-			X			-	X				X				X*	-	-	-	X	-			X*			
82			X-				X			-	-	X	-	*		-	X	-	*		X				X				X*	
83			X*	-			X		-		X				X				-	-	X*		-		-	X	*			X
84			-	X	-			X	-	-		X*					X	*			X		-			X*			-	X
85	X				X-				X*		-	-		X	-	*		X			-	X				-		X*		
86			X*				X				X*	-		-		X	-			X*	-	-		X					-	X
87			*	X	-			X				-	X	-				X*				X-	-	-		X*				X
88		X		-		X				X	-		*		X	-				X	-			-	X*	-				X*
89		X				X*					X*			-	X					X	-		*	-	X		-	-	X	-
90	X			-	X-				X		-	*	X				-	X*	-	-			X*				X			

REFERÊNCIAS BIBLIOGRÁFICAS

1. BALAS, Egon. *An Additive Algorithm for Solving Linear Programs with Zero-One Variables*. **Operations Research**, v. 13, p. 527-545, 1965.

2. BALAS, Egon; CERIA, Sebastián; CORNUÉJOLS, Gérard. *Mixed 0-1 Programming by Lift-and-Project in a Branch-and-cut Framework*. **Management Science**, Maryland, v. 42, n. 9, p. 1229-1246, setembro/1996.

3. BARBOZA, Angela Olandoski. **Aplicação de Algumas Técnicas da Pesquisa Operacional na Otimização de Horários de Atendentes em Central Telefônica.** Curitiba, 2000. Dissertação (Mestrado em Ciências) – Setor de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná.

4. BELLMAN, R. E.. **Dynamic Programming**. Princeton University Press, Princeton, (1957).

5. BRASIL. Decreto no 3.466, de 17 de maio de 2000. Aprova a Estrutura Regimental e os Quadros Demonstrativos dos Cargos em Comissão do Grupo-Direção e Assessoramento Superiores - DAS, das Gratificações de Exercício em Cargo de Confiança, das Gratificações de Representação pelo Exercício de Função e das Gratificações de Representação - GR do Ministério da Defesa, e dá outras providências. Disponível em: <http://www.planalto.gov.br>. Acesso em março de 2002.

6. BRASIL. Decreto no. 1145 de 20 de maio de 1994. Dispõe sobre os Quadros do Corpo de Oficiais da Ativa da Aeronáutica. Disponível em: <http://www.planalto.gov.br>. Acesso em fevereiro de 2002.

7. BRASIL. Lei n. 6880, de 9 de dezembro de 1980. Dispõe sobre o Estatuto dos Militares. Disponível em: <http://www.soleis.adv.br/ >. Acesso em fevereiro de 2002.

8. BROGGIO, Giancalo; PAOLETTI, Beniamino. *Operations Research in Industry – The Alitalia Experience*. ALITALIA Operations Research Department. **Le Bulletin**, v. 6, 1-3, primavera/verão, 2001.

9. BRONSON, Richard. **Pesquisa Operacional.** Coleção Schaum, Mc Graw-Hill, 1985.

10. CHRISTOFIDES, N. et al. **Combinatorial Optimization**. JohnWiley & Sons, Chichester, 1979.

11. CHUN, H.W. **Constraint Programming in Java with Jsolver**. *In*: *Proceedings of the First International Conference and Exhibition on the Practical Application of Constraint Technologies and Logic Programming*. Londres, 1999.

12. CHUN, H.W.; Chan H.C. et. al. **Nurse Rostering at the Hospital Authority of Hong Kong**. *In*: 2000 *Conference on Innovative Applications of Artificial Intelligence*, Austin, 2000.

13. CONSTANTINO, Ademir Aparecido. **Otimização de Escala de Trabalho para Condutores de Trem: Sequenciamento de Tarefas e Alocação Baseada em Preferência Declarada**. Florianópolis, 1997. Tese (Doutorado em Engenharia) - Departamento de Engenharia de Produção e Sistema, Universidade Federal de Santa Catarina.

14. DAKIN, R. J. *A Tree-Search Algorithm for Mixed Integer Programming Problems.* **Computer Journal**, V. 8, 250-255, 1965.

15. DUDZINSKI, K.; WALUKIEWICZ, S. **Exact Methods for the Knapsack Problem and its Generalizations**. EJOR 28, 3-21, 1987.

16. EXÉRCITO BRASILEIRO. **Braço Forte, Mão Amiga.** Disponível em <http://www.exército.gov.br>. Acesso em fevereiro de 2002.

17. GALLO, G.; HAMMER, P.L.; SIMEONE, B. **Quadratic Knapsack Problem**. Math. Prog. 12, 132-149, 1980.

18. GOLDBARG, Marco César;LUNA; Enrique Pacca L. **Otimização Combinatória e Programação Linear – Modelos e Algoritmos**. Rio de Janeiro, Campos, 2000.

19. HAJEK, B. *Extremal Splitting of Points Processes.* **Mathematics of Operations Research**, v. 10: 543-556, 1985.

20. HOFFMAN, K.L.; PADBERG, A. **Techniques for Improving the LP Representation of Zero-one Linear Programming Problems**. ORSA J. Computing, 3, 1991, 121-134.

21. KOOLE, Ger; SLUIS, Erik van der. *Optimal Shift Scheduling with a Global Service Level Constraint*. **Queueing Systems** , 28:337-347, 1998.

22. KUSUMOTO, S. **Nurse Scheduling System Using ILOG Solver**. *In*: *Proceedings of the Second ILOG Solver and Scheduler Users Conference*. ILOG, Paris, 1996.

23. LAND, A. H.; DOIG, A. G. *An Automatic Method of Solving Discret Programming Problems*. **Econometrica**, 28:3, 1960, 497-520.

24. LAU, H.C.; LUA, S.C. **Efficient Multi-Skill Crew Rostering via Constrained Sets**. *In*: *Proceedings of the Second ILOG Solver and Scheduler Users Conference*. ILOG, Paris 1997.

25. LAZARO, J.M.; ARISTONDO, P. **Using Solver for Nurse Scheduling**. *In*: *Proceedings of the First ILOG Solver and Scheduler Users Conference*. ILOG, Paris, 1995.

26. Manual do LINGO. **LINDO Systems, Inc**. N. Dayton, Chicago, Illinois. Outubro, 1999.

27. MARINHA DO BRASIL. **Bem Vindo a Bordo**. Disponível em: <http://www.mar.mil.br/brmar.htm>. Acesso em fevereiro de 2002.

28. MASON, Andrew J.; NIELSEN, David. *PETRA : A Programmable Optimisation Engine and Toolbox for Personnel Rostering Applications In*: **15th Triennial International Federation of Operational Research Societies Conference**. Agosto 16-20, 1999.

29. MAYERLE, Sérgio Fernando. **Programação Linear Inteira e Mista Programação Linear Binária**. Disponível em: <http://www.eps.ufsc.br/>. Acesso em fevereiro de 2002.

30. MITCHELL, John E. **Branch-and-Cut Algorithms for Integer Programming**. AMS, 1991. Disponível em: <http://www.rpi.edu/~mitchj/papers.html>. Acesso em fevereiro de 2002.

31. PADBERG, M.; RINALDI, G. **A Branch-and-Cut Algorithm for the Resolution of Large-scale Symmetric Traveling Salesman Problems.** SIAM Review, 33, 60-100, 1991.

32. POPOVA, Elmira; MORTON, David. *Adaptive stochastic manpower scheduling.* **Proceedings of the 1998 Winter Simulation Conference**, 661-668, 1998.

33. PUGET, J. F. **A C++ Implementation of CLP**. *In*: *ILOG Solver Collected Papers.* ILOG SA, França, 1994.

34. SHAMBLIN, James E.; STEVENS JR. **Pesquisa Operacional – Uma Abordagem Básica**. São Paulo, Atlas, 1989.

35. SIQUEIRA, Paulo Henrique. **Aplicação do Algoritmo do Matching no Problema da Construção de Escalas de Motoristas e Cobradores de Ônibus.** Curitiba, 1999. Dissertação (Mestrado em Ciências) – Setor de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná.