UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR DE ECONOMIA E GESTÃO

MESTRADO EM: MATEMÁTICA APLICADA À ECONOMIA E À GESTÃO

PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS NUMA MÁQUINA

APLICAÇÃO À DIVISÃO DE CORREIO

Jacinto Maurício Pires Nunes

Orientação: Professora Doutora Maria Teresa Chaves de Almeida

Júri:

Presidente: Professora Doutora Maria Teresa Chaves de Almeida

Vogais: Professora Doutora Margarida Maria Vaz Pato

Professor Doutor Miguel Fragoso Constantino

Setembro/97

UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR DE ECONOMIA E GESTÃO

MESTRADO EM: MATEMÁTICA APLICADA À ECONOMIA E À GESTÃO

PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS NUMA MÁQUINA

APLICAÇÃO À DIVISÃO DE CORREIO

Jacinto Maurício Pires Nunes

Orientação: Professora Doutora Maria Teresa Chaves de Almeida

Júri:

Presidente: Professora Doutora Maria Teresa Chaves de Almeida

Vogais: Professora Doutora Margarida Maria Vaz Pato

Professor Doutor Miguel Fragoso Constantino

Setembro/97

À minha mãe

À Dulce

There are certain things about swinging a cat by the tail
that you can only learn by swinging a cat by the tail.

Mark Twain, citado por João Lobo Antunes, Um modo de ser

Para bom entendedor,
meia palavra basta.

Anónimo (?)

α ..... Sistema de processamento (*machine environment*)

β ..... Características das tarefas a processar em α

γ ..... Critério de optimalidade para o escalonamento

$\alpha \mid \beta \mid \gamma$ ..... Classe do escalonamento

Δ_t ..... Duração de um período em que se subdivide o tempo total de funcionamento do sistema de processamento.

Ω ..... Quando o problema do escalonamento é modelizado como de optimização de rotas, representa o depósito

CDP ..... Centro de Distribuição Postal

c_{ds} ..... Quando o problema do escalonamento é modelizado como de afectação, representa o custo de afectar o lote d a um subperíodo s

c_f ..... Quando o problema do escalonamento é modelizado como de afectação, representa o custo de afectar um lote fictício a qualquer subperíodo s (fictício ou não)

C_k ..... Instante de conclusão (*completion time*) da tarefa k

$C_k(S)$ ..... Momento de conclusão da tarefa k segundo um escalonamento específico, S

c_{min} ..... Quando o problema do escalonamento é modelizado como de afectação, representa o custo mínimo de afectar um lote a qualquer subperíodo s

CTC ..... Centro de Tratamento de Correio

D ..... Número de cartas por lote

d ..... Quando o problema do escalonamento é modelizado como de afectação, simboliza um lote a afectar a um subperíodo s

Dist ..... Quando o problema do escalonamento é modelizado como de optimização de rotas, representa a "distância" entre o depósito e uma saída dos *OCR* ou entre duas saídas dos *OCR*

Divisora ..... Máquina de triagem automática de cartas por destinos

d_k ..... Instante em que a tarefa k tem que estar concluída (*due time*)

Giro ..... Percurso que um carteiro efectua diariamente para entrega de correio

Hora de corte ..... Momento até ao qual as cartas (correspondências), têm que estar processadas num CTC

i ..... Período genérico, correspondendo a uma subdivisão do tempo total de funcionamento do sistema de processamento (i = 1, 2, ..., L)

i_l ..... Quando o problema do escalonamento é modelizado como de afectação, representa uma subdivisão do período i (l =1, 2,..., K)

Indexação ..... Colocação de barras flourescentes nas cartas para que possam ser triadas automaticamente

Janela temporal ..... Quando o problema do escalonamento é modelizado como de optimização de rotas, representa o intervalo de tempo entre a chegada (r_k) e hora de corte (d_k) (*time window service*)

J_k ..... Tarefa (*job*) k

K ..... Quando o problema do escalonamento é modelizado como de afectação, corresponde ao número de subperíodos do período i

L ..... Número de períodos em que funciona o sistema de processamento. (Cobre todo o tempo de laboração de um CTC)

M_i ..... Máquina i, integrante do sistema de processamento α

n_T ..... Número de tarefas concluídas com atraso

$n_T(S)$ ..... Número de tarefas concluídas com atraso a partir do escalonamento específico *S*

OCR ..... Leitor óptico (*Optical Character Reader*)

O_{ik} ..... Operação executada na máquina i correspondente à tarefa k

p_{ik} ..... Tempo de execução da tarefa k na máquina i

Prioridade ..... Importância relativa atribuída a um destino (ponderador)

Q_i^j ..... Quantidade de cartas "geradas" na saída j dos *OCR* durante o período i

r_k ..... Instante a partir do qual a tarefa k pode ser processada (*ready time*)

S ..... Escalonamento (*schedule*)

s ..... Quando o problema do escalonamento é modelizado como de afectação, representa uma subdivisão do período i (i_l=s; l =1, 2, ..., K)

S_j ..... Saída j dos *OCR*; "porta" atribuída a um grupo de destinos, onde são recolhidas as cartas após a indexação automática

Tempo de esvaziamento ou de preparação ..... Tempo necessário para efectuar a mudança de alimentação da divisora (substituição da saída j dos *OCR* pela saída m) (*set-up time*)

$T_j(i)$ ..... Tráfego "gerado" na saída j dos *OCR* até ao início do período i e ainda não triado até final de i

Triagem ..... Divisão/separação de cartas por destinos

U_k ..... Indicador de atraso (*unit penalty*) da tarefa k; $U_k = 0$ se J_k concluída em tempo; $U_k = 1$ caso contrário

V_i ..... Velocidade de processamento da máquina (divisora) i

v_{jq} ..... Quando o problema do escalonamento é modelizado como de optimização de rotas, representa a "velocidade de circulação" entre as saídas j e q dos *OCR*

w_k ..... Ponderador da tarefa k no conjunto de todas as tarefas a processar (1, 2, ..., n)

RESUMO

"- Lembro ao nobre deputado que
a câmara não é aula de retórica"

Camilo Castelo Branco, A queda de um anjo

Nesta dissertação é estudado um problema de escalonamento de tarefas num sistema de processamento onde máquinas de divisão de correio são os processadores.

Uma tarefa (*job*) consiste na triagem automática de um lote de cartas, sendo objectivo minimizar o número de lotes recebidos e não processados em cada dia.

Os lotes, formados por cartas para vários destinos que importa individualizar, são triados em Centros de Tratamento de Correio (CTC), com uma só máquina ou com duas máquinas em paralelo.

Embora a quantidade diária de objectos chegados a cada CTC seja aleatória, o número de lotes a processar é assumido como pré-determinado (valores médios estimados), assim como igualmente conhecidos são os *ready times*, *due times*, tempos de execução e prioridade de tratamento (ponderador - que pode ser igual ou diferenciado) de cada lote. Necessário considerar serão ainda tempos de *set-up* das divisoras, em função da sequências de lotes.

A oportunidade deste trabalho resulta da reformulação de processos e consequente reequipamento que decorre na empresa CTT - Correios de Portugal, SA.

Começa-se por uma descrição das principais operações num CTC, para poder definir com precisão o problema em análise.

É feita depois uma revisão de literatura existente sobre problemas de escalonamento, essencialmente, com uma só máquina.

Realiza-se em seguida um enquadramento do problema presente com os anteriormente estudados e referidos. Perante a não total satisfação com os métodos encontrados para a resolução deste caso, propõe-se e argumenta-se a validade de um algoritmo para gerar o escalonamento na condição de existir uma única máquina de divisão, que minimiza o número de lotes não processados na hipótese de tempos de *set-up* todos nulos.

Dado um CTC poder ter duas máquinas de divisão em paralelo, sugere-se e defende-se uma forma de escalonamento para as duas máquinas com base num escalonamento obtido para apenas uma.

Dois outros capítulos são dedicados a modelizar este problema de escalonamento como, respectivamente, um problema de geração de rotas de veículos, que cobre a hipótese de *set-up* dependente da sequência, e um problema de afectação.

Seguem-se-lhes os resultados computacionais e, a finalizar, as conclusões.

Palavras chave: Problemas de Escalonamento de Tarefas; Problemas de Optimização de Rotas; Problemas de Afectação; Heurísticas.

São tão breves os pensamentos
e tão longas as palavras.

Lídia Jorge, O cais das merendas

Uma imagem vale mil palavras

Provérbio chinês

Saravah!

Vinicius de Moraes, Samba da Benção

A minha primeira manifestação de gratidão é dirigida à Professora Doutora Maria Teresa Chaves de Almeida.

Não me guiam rituais ou circunstancialismos ao deixar aqui expresso o meu reconhecimento pela sua orientação competente, motivante e dedicada desta dissertação.

Além deste apoio determinante, são-me devidos outros agradecimentos pela colaboração influente na tarefa de realização deste trabalho:

À minha mulher, Dulce, pela compreensão e incentivo.

Aos meus familiares e amigos mais próximos pelo estímulo, que também me quiseram e souberam dar.

Ao Dr. Carlos Carneiro da Optimiza, pelo seu interesse amigo e ajuda na utilização do *software* Optrak.

À Dra. Maria Conceição Fonseca da Faculdade de Ciências da Universidade de Lisboa, pela cedência da aplicação para tratar problemas de afectação e instruções para o seu uso.

Ao meu colega nos CTT Eng. Albano Rosa, pelos comentários e revisão do texto.

À Dra. Maria Filomena Santos, pela leitura crítica da dissertação.

Ao também meu colega nos CTT, Eng. José Pedro Rufino, pela execução de algumas figuras.

Ao Eng. José Manuel Coutinho, director dos CTT, pela sugestão e oportunidade do tema.

Aos CTT, na pessoa do Dr. Alberto Pimenta, pelas facilidades concedidas para realização do mestrado.

Contudo, eventuais erros ou omissões que ainda possam persistir nesta dissertação, são da exclusiva responsabilidade do seu autor.

CAPÍTULO 1

INTRODUÇÃO

(...) não há um princípio para as coisas e para as pessoas,
tudo o que um dia começou tinha começado antes (...)

José Saramago, A jangada de pedra

É objectivo desta dissertação contribuir para a resolução de um problema operacional colocado aos CTT - Correios de Portugal, SA, especificamente no seu subsistema de tratamento automático de correspondências.

O processamento mecanizado presente possibilita a separação de correspondências (categoria de objectos constituída na quase totalidade pelas "cartas") até um nível de aproximadamente 400 destinos, os chamados Centros de Distribuição Postal (CDP), com limites geográficos inspirados na delimitação dos concelhos do país e identificados por 4 dígitos - o actual código postal.

O sistema operativo postal, que será descrito com maior pormenor no capítulo seguinte, pode decompor-se, no fundamental para este trabalho, nas seguintes actividades: aceitação, triagem por destinos, transporte e distribuição.

A aceitação corresponde à recolha dos objectos recebidos dentro das fronteiras de cada CDP, os quais são enviados, na totalidade ou na maior parte, para centros agregadores, chamados Centros de Tratamento de Correio (CTC), onde funcionam equipamentos de divisão automática de correspondências, que efectuam a triagem intensiva por código postal.

Finalizada esta operação os objectos são transportados para os CDP onde habitam os destinatários, para serem distribuídos pelos carteiros.

É precisamente aqui, na forma como chegam os objectos aos CDP vindos dos CTC, que reside uma oportunidade de incrementar, substancialmente, a produtividade da cadeia de tratamento.

Os carteiros, de modo a que possam fazer a entrega porta-a-porta, necessitam de ter o correio separado por itinerários fixos, os chamados "giros", que realizam diariamente.

Esta operação tem sido integralmente manual, já que a divisão automática só está a desagregar até ao código de CDP, mas é possível com novos equipamentos levar a divisão ao nível do "giro" para os CDP de maiores dimensões, equivalendo a passar dos presentes 400 para perto de 3000 códigos de destino (que incluem CDP e cerca de 45% de giros de todo o país).

Para poder viabilizar esta maior fragmentação, o código postal será ampliado de 4 para 7 dígitos, mas enquanto não estiver suficientemente divulgado e enraizado, vai ter que ser lido o próprio endereço para identificar o seu código de destino.

As máquinas de divisão de correio existentes carecem de um código de barras onde é representado o código postal, legível nas correspondências, para que possam direccionar diferentes códigos para diferentes saídas.

A colocação destas barras tem vindo a ser efectuada por operadores cuja função é ler o código postal do destinatário e digitá-lo num teclado, produzindo a sua impressão no canto inferior direito das cartas.

Ora se só com a extensão do código postal para 7 dígitos a produtividade do processamento automático já era negativamente afectada pois, ou se duplicava o número de operadores, ou se reduzia para metade o ritmo de colocação de códigos (com aumento, previsível, de erros de digitação), a situação seria insustentável se o endereço tivesse que ser lido e o código postal determinado pelos operadores.

Surge, então como óbvia, a necessidade de aquisição de leitores ópticos (*OCR - Optical Character Reader*), que consigam ler os endereços e os exprimam num código de barras interpretável pelas divisoras; o que os CTT já efectuaram e instalaram nos CTC em meados de 1996 para testes que se vão estender até meados de 1997.

As divisoras automáticas existentes nos CTC são equipamentos computorizados que simultaneamente podem triar os objectos por um máximo de 220 saídas.

Ora para que uma carta não seja dividida mais que uma vez, é imperioso que haja uma separação prévia à passagem das cartas pelas máquinas de divisão, mesmo com 400 destinos, que se reforçará com os pretendidos 3000.

Isto significa que terão de se formar grupos de códigos de destino (correspondendo a uma pré-divisão), sendo cada grupo associado a uma saída dos *OCR* (em qualquer saída dos *OCR* não pode haver mais que 220 destinos diferentes), que formarão lotes para "passar" nos equipamentos de divisão; lotes estes, que não são todos constituídos e enviados para dividir no mesmo instante, mas num "caudal" de intensidade dependente do ritmo de chegada de cartas ao CTC e devendo ser triados até um momento pré-estabelecido (função da distância que separa o centro de tratamento dos CDP a que se destinam).

Embora as quantidades de tráfego postal recebidas diariamente em cada CTC sejam aleatórias e desconhecidas *a priori*, são assumidas como valores médios, previamente estimados.

Decorrem, daqui, dois problemas facilmente detectáveis:

- Quantos grupos de destinos formar (o que significa determinar o número de saídas mínimo dos *OCR*), e com que códigos postais;

- Pretendendo que as cartas sejam divididas numa janela temporal definida, *grosso modo*, pelo momento de chegada ao CTC e a hora limite para serem transportadas para o CDP de destino, como escalonar nas divisoras os lotes provenientes de cada saída dos *OCR*, de forma a maximizar o número das correspondências processados sem atraso, o que para os CTT se exprime em procurar tratar o máximo de objectos até ao dia seguinte ao da sua aceitação.

  Elevando o grau de dificuldade, pode ainda tornar-se necessário considerar que os destinos têm prioridades diferenciadas e que a mudança de lotes nas máquinas de divisão quando se alternam grupos de códigos distintos pode levar, como se justificará, à inclusão dos chamados tempos de preparação (*set-up*).

Será do segundo problema que este estudo se vai ocupar.

Começa-se por apresentar, no Capítulo 2 a empresa CTT - Correios de Portugal, SA nas componentes sócio-económica e estrutura operacional. Evidenciam-se em seguida aspectos de evolução tecnológica e descreve-se o essencial das operações num CTC para chegar à compreensão e formalização do problema que se vai analisar.

O Capítulo 3 dedica-se a problemas de escalonamento em geral, com relevo para sistemas de processamento com uma só máquina, procurando relacionar o caso presente com classes de problemas descritas na literatura, tendo o propósito de conceber uma abordagem mais formalizada e menos empírica. Encerra este capítulo, com o mesmo intento anterior, a referência a métodos de resolução por numeração implícita, melhorativos e construtivos.

Configurado o campo dos problemas de escalonamento, é preocupação do Capítulo 4 situar este caso particular e propor e fundamentar um algoritmo, construtivo, que o resolva na versão base (sem admitir tempos de preparação das divisoras e com priori-

dades iguais). O algoritmo apresentado aplica-se directamente aos CTC com uma só máquina de divisão (Coimbra e Porto), mas mostra-se que é facilmente adaptado para as duas máquinas divisoras instaladas no CTC de Lisboa. Complementarmente, considera-se a possibilidade de um escalonamento onde os destinos não têm todos as mesmas prioridades.

Quando se consideram nas divisoras tempos de preparação (*set-up*), função da sucessão de lotes cai-se em escalonamentos referenciados como de tempos de processamento dependentes da sequência (*sequence-dependent processing times*), afins aos problemas do caixeiro viajante e suas extensões.

O Capítulo 5 enfrenta esta hipótese tratando o problema completo como um *vehicle routing problem*, dispondo-se de uma aplicação comercial - Optrak - para a tarefa computacional.

O Capítulo 6 é preenchido com a concepção de um escalonamento integrado de todas as divisoras, para avaliar benefícios de permuta de correio inter-CTC, modelizando o segundo problema exposto com ou sem prioridades (mas sem tempos de preparação ou esvaziamento), como um problema de afectação.

O Capítulo 7 exibe os resultados obtidos e por último, no Capítulo 8, extraem-se as conclusões do estudo.

CAPÍTULO 2

OS CTT E A DISTRIBUIÇÃO DE CORREIO: O PRESENTE E O FUTURO PRÓXIMO

Abri o peito e mostrei-lhe
a areia, a pedra britada,
os planos da grande estrada
onde o Anjo se ajoelhe

António Gedeão, Poesias completas

2.1.Aspectos sócio-económicos

"Correio" é um termo reconhecido e associado com comunicação de mensagens e entrega de mercadorias, pela quase totalidade da população portuguesa.

A prestação de serviços contidos no entendimento abrangente de "correio", sintetizado em: transportar objectos, estabelecendo ligação entre um remetente e um destinatário, é assegurada pela empresa CTT - Correios de Portugal, SA.

Esta entidade é a herdeira dos Correios Públicos, criados por D. Manuel I em 1520, vendidos em 1606 a um particular por Filipe II e reintegrados na Coroa em 1797, por D. João VI [Cardoso, 1984]. Em 1969 deixaram de ser geridos por um Correio-Mor passando a empresa pública, e em 1992 separaram-se os sectores telecomunicações e correio, tornando-se uma sociedade anónima de capitais exclusivamente públicos.

Os CTT operam em quatro grandes áreas de negócios: correspondências, encomendas, serviços *courier (EMS - Express Mail Service)* e serviços financeiros. Decorrente da sua actividade básica, actuam ainda no domínio da filatelia/coleccionismo.

No âmbito da sua actuação no mercado, os CTT apenas têm o monopólio legal, mas não efectivo, no sector das correspondências, o qual tende a ser progressivamente libe-

ralizado, permitindo a entrada de novos operadores nos segmentos de maior rendibilidade (urbano, correio rápido, etc), [Pilar, 1996].

A avaliação expedita do seu peso na economia nacional pode fazer-se pelo volume de emprego (cerca de 16000 pessoas), pelos resultados (em 1995, 80.5 e 76 milhões de contos, respectivamente, em proveitos e custos operacionais), com o número de objectos movimentados ao longo do ano (um pouco além de mil milhões) e pela existência de cerca de mil estações de correio espalhadas por todo o país [CTT, 1996].

As discussões, no contexto da União Europeia, com vista à abertura de mercados na actividade postal e as preocupações de gestão empresarial trazidas pelo novo estatuto, obrigam os CTT a encontrar soluções para melhorar a produtividade e a qualidade de serviço que assegurem uma empresa saudável.

2.2.Estrutura operacional

Embora a actividade "correios" seja sugestiva e se percepcione o fundamental das suas tarefas, é indispensável algum detalhe do seu sistema operativo, para entendimento correcto dos problemas que se pretendem expor e avaliar as soluções preconizadas.

Do ponto de vista da actividade postal, Portugal está dividido, a um primeiro nível, em áreas geográficas, com fronteiras induzidas pela divisão administrativa dos concelhos, cujo centróide é o local de edificação do chamado Centro de Distribuição Postal[1], início e fim (com poucas excepções), do trajecto interno dos objectos confiados aos CTT.

[1] Nas maiores cidades existem vários CDP

É da sede do CDP que partem os carteiros para efectuarem os "giros", isto é, os percursos diários de entrega de correio porta-a-porta aos destinatários. Inversamente, na recolha, os objectos recebidos dentro dos limites de um CDP, são para lá conduzidos como primeira etapa do processo "produtivo".

Agregando conjuntos de CDP, existe um segundo nível da rede postal, menos visível para o utente, formado pelos Centros de Tratamento de Correio, onde é realizada a divisão massiva dos objectos originados nos CDP da sua área de influência, e que os expede para os CDP de destino.

O processo postal, no que se reveste de importância para este trabalho, pode ser esquematizado nas funções [Rosa, 1990]:

(i) Aceitação;

(ii)Triagem por destinos;

(iii)Transporte;

(iv)Distribuição.

O significado a atribuir a cada um dos itens anteriores vem no que se segue:

(i) A aceitação, é a fase de recolha de objectos, efectuada na área de influência de cada CDP (origem), depositados em marcos e caixas, aos balcões das estações, ou recebidos em casa do cliente.

(ii)A triagem, pode começar a fazer-se, manualmente, nos CDP origem com a disjunção dos que têm remetente e destinatário no próprio CDP, com os que têm destino noutro CDP e que são enviados para tratamento no CTC a que o CDP origem está ligado.

Nos CTC executa-se, essencialmente, a divisão de grandes volumes de tráfego; mas podem ser também pontos de trânsito para concentração e reenvio para outros CTC (onde se desenrolará o processo de divisão).

(iii)Os transportes têm como função ligar os CDP aos CTC no envio de objectos a tratar, ou o caminho inverso quando estes estão já triados. Suportam igualmente a movimentação inter-CTC e, nas cidades onde estão implantados CTC, fazem ligações directas, entre estações, marcos, caixas e grandes clientes, e respectivos Centros de Tratamento de Correio.

São assegurados por uma rede rodoviária com itinerários e horários fixos, em frota em grande parte própria.

(iv)A distribuição é o último elo na cadeia e a conclusão do processo de " levar carta a Garcia". Inicia-se quando concluída a separação por destino e os objectos já estão no respectivo CDP. Antes da saída para a distribuição propriamente dita, é necessário que estejam colocados no "giro" que os vai entregar e numa sequência, em cada artéria, que minimize o percurso dos carteiros.

2.3.Evolução tecnológica

Até 1978 todo o processamento de divisão dos objectos postais era inteiramente manual.

Nesse ano, foi introduzido um código postal de 4 dígitos ("Código postal - meio caminho andado"), identificando as áreas postais já referidas como CDP. Passou a figurar como elemento fundamental do endereço, para permitir uma reformulação da função distribuição associada ao tratamento mecanizado.

A intenção da sua criação foi viabilizar o projecto de divisão automática, que previa e concretizou a construção de 3 Centros de Tratamento de Correio - Lisboa, Coimbra e Porto - onde foram instalados equipamentos para processamento das correspondências.

Os sistemas então disponíveis que foram adquiridos e que se mantêm activos, exigem uma prévia codificação, a realizar "manualmente" (dado o estado tecnológico, à data da sua compra), por operadores cuja função consiste em ler o código postal correspondente ao CDP do destinatário e digitá-lo num teclado. Como resultado, obtém-se um código de barras colocado no canto inferior direito das correspondências legível pelas máquinas divisoras que possibilita a triagem por essa unidade postal.

Dado este nível de separação ser ainda demasiado agregado, o tratamento tem que ser prosseguido com o correio após transporte para o CDP destino, sujeito a operações manuais de divisão e sequenciamento até estar em condições de entrega, pelos carteiros, aos destinatários.

Estas tarefas de separação manual levam a classificação do correio até subdivisões do CDP nas suas "unidades constituintes" chamadas "giros", que, repete-se, são conjuntos de itinerários a realizar diariamente (e constantes) que os carteiros efectuam para entrega de objectos porta-a-porta.

A continuada investigação na procura de maior grau de automação nos processos de tratamento, permitiu o aparecimento de leitores ópticos de endereços que conseguem ler caracteres manuscritos e "dactilografados" (estes com maior taxa de sucesso) e fazer a indexação automática, isto é, substituem os operadores na tarefa de colocação do código de barras, que representa o destino, sobre os objectos, preparando-os para a divisão mecanizada.

Seguindo a tendência dos países tecnologicamente mais avançados, os CTT instalaram este tipo de sistemas e iniciaram a fase de testes, a partir de meados de 1996, nos três CTC já enumerados, com o objectivo de melhorar, substancialmente, qualidade e produtividade.

Embora, como se disse, os *OCR* possam ler a totalidade do endereço (e fá-lo-ão numa primeira fase), a taxa de rejeição é bastante reduzida se o conjunto dos caracteres possíveis não for extenso, já que diminui o leque das interpretações verosímeis.

O aliar o máximo desempenho dos *OCR* com a intenção de levar a divisão mecânica tão longe quanto o permitido, conduzem à extensão do actual código de 4 para 7 dígitos, a ser lido e indexado de modo automático, viabilizando a utilização das divisoras na triagem correspondente aos referidos "giros". Levar-se-à, assim, a classificação automática dos objectos a um nível muito mais desagregado (até ao "giro"), que anteriormente (de cerca de 400 para 3000 destinos). A eliminação nos CDP destino de maior dimensão, do trabalho actual de divisão por "giro" (mantendo-se ainda o de ordenar dentro de cada artéria)[2], resultará num ganho substancial de produtividade.

[2] A médio prazo, prevê-se a introdução de equipamentos para fazer o sequenciamento dos objectos dentro de cada giro.

2.4.Problemas emergentes

Nas soluções procuradas para incremento da competitividade estão, naturalmente, saltos tecnológicos como os referidos, os quais levantam, eles próprios, a necessidade de adequações à realidade onde são integrados.

O enquadramento e a explicitação de alguns problemas, resultantes do funcionamento dos *OCR* , far-se-ão nos próximos pontos.

2.4.1.Descrição sumária das operações num CTC

Para os propósitos deste estudo, as operações dentro de um CTC que asseguram a triagem por destinos finais das correspondências esquematizam-se no que se segue:

- Recepção dos objectos

  A maior concentração de tráfego, encaminhado para um CTC e originado em CDP da sua área de influência, ocorre entre as 17.30 e as 20.30 horas. No entanto, a chegada de objectos com proveniência directa de estações, marcos, caixas e clientes de grande dimensão (com várias tiragens ao longo do dia), e as ligações inter-CTC sustentam um fluxo quase contínuo.

- Preparação e separação por tipos

  Parte do correio, nomeadamente o depositado nos marcos e caixas, tem volumetria e espessura diversa, isto é, recebe-se uma mistura de objectos "finos", "médios" e "grossos".

Para isolar cada categoria, existem equipamentos mecânicos, que se podem descrever como cilindros rotativos com ranhuras actuando como "filtros", que deixam, ou não, passar cada um dos tipos enumerados.

Obtém-se daqui a classe "finos" (genericamente "cartas"), individualizada das restantes. Uma pequena curiosidade que se acrescenta, é que é também função destas máquinas deixar as "cartas faceadas" (à saída todas as cartas têm a mesma orientação em relação ao endereço e posição do selo) e obliteradas (com o selo carimbado).

- Separação automática de correspondências

  - Indexação

    Ultrapassadas as duas etapas anteriores, e não dando relevo a uma passagem que selecciona dentro dos "finos" os que podem ser processados mecanicamente, atinge-se a indexação.

    Basicamente, esta operação descreve-se como a tradução de um "destino" num código de barras verticais a colocar no canto inferior direito das correspondências.

    Com a entrada em funcionamento dos *OCR* é a chamada indexação que sofre uma alteração qualitativa, embora a tarefa não se transforme em totalmente automática. Objectos que sejam rejeitados na passagem dos *OCR*, ou que previamente se detecte não reunirem condições para serem lidos automaticamente, são conduzidos para postos de indexação manual.

    A diferença entre estes dois modos de colocação de barras é: no manual, o código apenso representa apenas o CDP destino, enquanto que com os *OCR* além do CDP pode também ser escrito o código de "giro".

Como resultado, obtêm-se dois grupos de objectos, onde num se pode levar a sua classificação automática até muito maior individualização, ao passo que no outro há maior exigência de recursos humanos para conseguir a mesma diferenciação (quando as cartas chegarem ao CDP destino).

Numa primeira fase os *OCR* lerão o próprio endereço pesquisando-o num ficheiro de endereços que contém, também, o código de destino correspondente. Quando o encontram, são escritas as respectivas barras; caso contrário está-se num exemplo de rejeição.

Quando estiver suficientemente divulgado o novo código postal (CDP+GIRO), de 4+3 dígitos ("Código postal - cada vez mais perto"), passará a ser este o objecto de leitura, garantindo-se, como se disse, maior taxa de indexação automática.

Nas indexações manual e automática existe simultaneamente uma pré-divisão que distribui os objectos por diversas saídas e cuja utilidade se pormenorizará adiante.

- Divisão

Feita a indexação, o correio está apto a ser tratado nos equipamentos de divisão mecânica.

As divisoras automáticas de correio existentes são equipamentos igualmente computorizados com um máximo de 220 saídas. Lêem o código de barras correspondente ao "código destino" e comparam-no com os registos de um ficheiro carregado na sua memória *RAM*. Encontram assim, a "porta de saída" da máquina para onde devem deslocar os objectos.

Para considerações posteriores, importa referir dois detalhes operacionais:

. Dado o limite de saídas das divisoras, o número de CDP (cerca de 400) e a decisão por razões de gestão, de que os objectos só passam uma vez pelas divisoras, então tem que existir prévia triagem para agrupar CDP. Cada grupo passará em períodos diferentes, com o ficheiro adequado em *RAM*.

Este último problema toma compreensivelmente maior acuidade, quando se pretende separar um número de destinos bastante superior.

A pré-divisão, relembra-se, faz-se em paralelo com a indexação; quer os *OCR*, quer a indexação manual (embora em menor escala) permitem, como referido, algum grau de separação de correio.

. Outro destaque vai para o tempo gasto na preparação de uma máquina divisora quando se muda a saída dos *OCR* que a alimenta.

Exemplificando, considere-se a divisão por "giro" de dois grupos de CDP numa máquina. A informação em memória *RAM* do equipamento para a divisão do primeiro grupo, é como se disse, diferente do segundo, pois terá que conduzir códigos diferentes para outra distribuição de saídas, correspondendo a destinos diversos. Além disso, existe a necessidade de tempos de espera entre o fim de um programa e o início do que lhe sucede (que talvez possa ser reduzida a um valor negligenciável), que se explica pela indispensabilidade de retirar todos os objectos divididos do primeiro grupo das respectivas saídas, para não se misturarem com os do seguinte.

- Expedição

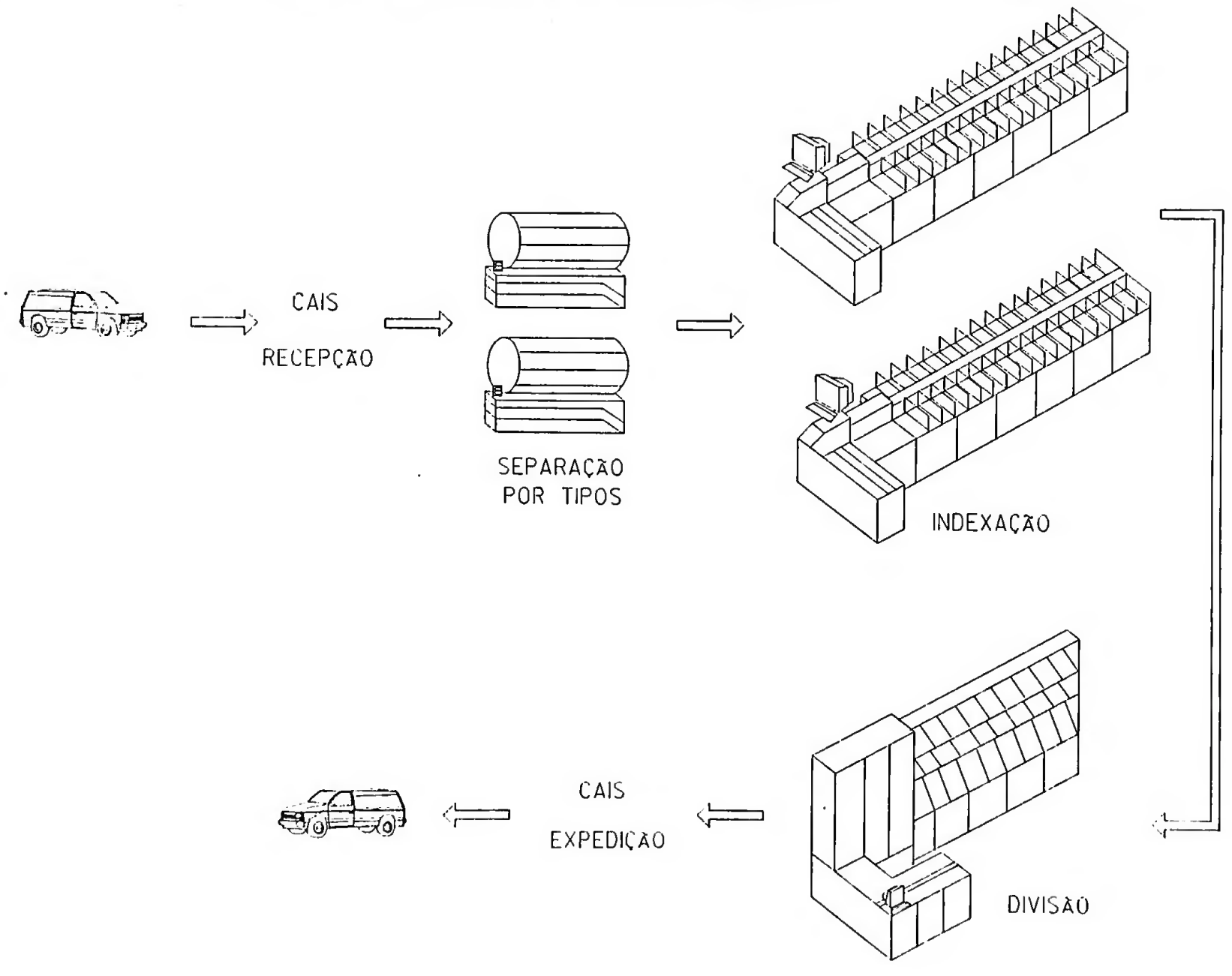
Completada a divisão, os objectos são encaminhados para o cais do CTC para serem levados para os vários CDP de destino.

O estabelecimento de horas de começo dos giros de distribuição, obriga a horas limite de partida dos veículos de transporte, e consequentemente, outro factor a ponderar adiante, "horas de corte" para finalização do processo de triagem.

A próxima figura é a imagem síntese da descrição anterior:

Figura 1

DIAGRAMA DE OPERAÇÕES NUM CTC



2.4.2.Formulação dos problemas

Decorrente do exposto no ponto anterior são facilmente identificáveis dois problemas a nível de cada CTC.

O primeiro prende-se com o estabelecimento dos agrupamentos de destinos nas saídas dos *OCR*; o segundo com o escalonamento destes agrupamentos de cartas que saem dos *OCR* e da indexação manual[3] e que terão que passar pelas divisoras.

2.4.2.1.Agrupamento de destinos nos *OCR*

Com o actual limite de saídas das máquinas de divisão, obviamente não podem ser separados, em cada momento, mais de 220 destinos.

Com a possibilidade de dividir automaticamente, não apenas por CDP mas ao nível do "giro", mesmo sem considerar todos os "giros" diários, pretende-se atingir um número de destinos finais na ordem dos 3000.

Com o pressuposto de que os objectos não "passam" mais do que uma vez nas divisoras, torna-se indispensável uma pré-triagem, em que se faça agrupamento de destinos, sujeito a determinadas condicionantes.

Um condicionamento, evidente, é a de que o número de destinos diferentes agrupados numa saída dos *OCR* não pode ser superior a 220.

Outra restrição respeita ao equilíbrio das "horas de corte" dos diversos destinos em cada grupo. Para que todos os destinos de um grupo sejam divididos até à sua hora

[3] Os objectos com origem na indexação manual podem ser adicionados a algumas saídas dos *OCR* ou formar novos grupos.

limite, é a daquele com menor "hora de corte", que definirá a de todo o conjunto. Se for grande a variância relativamente às "horas de corte" em cada saída dos *OCR*, poder-se-á diminuir a eficiência de todo o tratamento.

É um problema cuja solução é um compromisso entre a variabilidade das horas de corte dentro de cada grupo e o número de saídas disponíveis dos *OCR*.

Se a diferença imposta entre o máximo e o mínimo das horas de corte em cada saída dos *OCR* for pequena, então o processo de divisão é beneficiado, já que as horas limite de cada grupo[4] são muito próximas das horas limite efectivas de cada carta pertencente ao grupo e, portanto, nenhuma carta tem *due date* com antecipação significativa isto é, o período em que é possível processá-la será muito próximo do real. A contrapartida é o previsível aumento do número de saídas dos *OCR* (que no limite incluiria a própria divisão), com o encarecimento associado.

Aumentando a diferença entre os maior e menor valores das horas de corte dentro de cada agrupamento então, evidentemente, diminui-se o número necessário de saídas dos *OCR*, mas muitas cartas que poderiam ser tratadas sem atraso deixam de o ser, porque o menor valor dentro do grupo é significativamente inferior às horas limite respectivas, reduzindo o intervalo possível de tratamento.

É um problema que pode ser, em sentido lato, designado como de análise ou decisão de objectivos múltiplos e resolvido recorrendo à programação multiobjectivo.

Poderiam então associar-se à diferença máxima entre horas de corte dentro de qualquer grupo e ao número de número de saídas dos *OCR* metas a atingir e minimizar a soma ponderada dos seus desvios em relação a essas metas [Tavares et al, 1996] e [Guerreiro et al, 1985].

---

[4] Continuando a aceitar que é fixada pelo destino de menor hora de corte dentro do grupo.

A sua solução pode, no entanto, ser procurada, numa forma iterativa e mais simples, variando a amplitude máxima permitida das horas de corte em qualquer conjunto (que passaria a restrição), até que o número de saídas dos *OCR* mínimo, resultante, seja satisfatório. A verdadeira restrição em ambos os métodos a ter em conta, será, obviamente, o número de destinos diferentes em cada grupo não ultrapassar o número de saídas das máquinas de divisão.

2.4.2.2.Escalonamento nas máquinas de divisão

A existência de um limite, para o número de destinos que é possível separar em simultâneo, inferior ao desejado, obriga, como se disse, a um prévio agrupamento.
Esta agregação, resultado da solução do problema anterior e da junção dos objectos provenientes da indexação manual, vai possibilitar, fazendo a alimentação das divisoras grupo a grupo, a separação da totalidade de destinos.
É daqui sugestiva a necessidade de um escalonamento para a passagem nas divisoras dos diversos grupos, com dois factores determinantes na modelização do problema:

- O cumprimento das "horas de corte";
- A chegada de correio aos CTC e, consequentemente, à divisão.

Sendo relembra-se (2.4.1.), um processo quase contínuo (dentro das horas de funcionamento de um CTC), a decisão de "passar" um grupo, nas máquinas de divisão dentro de uma janela de tempo, é condicionada pela disponibilidade dos objectos.

Explicitados os aspectos que configuram o fundamental da forma de escalonamento, poderá, complementarmente, enriquecer-se o modelo com a introdução de dois novos parâmetros:

- Prioridades a atribuir a cada grupo;
- Tempo de preparação das divisoras para mudança de grupo (saída dos *OCR*).

Fazendo agora a conclusão do exposto, pode formular-se este problema no seguinte modo:

Organizar a alimentação das máquinas de divisão, isto é determinar a sequência e o período de passagem dos agrupamentos de cartas provenientes da indexação, de modo a minimizar o número das não processadas dentro das "horas de corte", com obediência a:

- Diagrama horário de cargas por destinos;
- Prioridades por destinos;
- Tempo de preparação das máquinas de divisão entre grupos.

Uma justificação de sentido prático é oportuna aqui, já que a hipótese de se considerarem valores médios de tráfego pode parecer paradoxal: Qual a utilidade de se procurar um escalonamento para as máquinas de divisão se é desconhecida previamente a quantidade de cartas que são recebidas em cada dia?

Esta aparente contradição tem uma resposta dupla:

Primeiro, o escalonamento encontrado para este problema particular, mais fácil de tratar com a simplificação de valores não estocásticos, pode sugerir regras aplicáveis para a situação de aleatoriedade de facto.

Em segundo lugar, é importante quantificar o limite de cartas que é possível processar em cada CTC, na situação actual e para futuros projectos de aquisição de máquinas divisoras.

CAPÍTULO 3

PROBLEMAS DE ESCALONAMENTO DE TAREFAS

Nada é tão prático como uma boa teoria

Kurt Lewin

3.1.Introdução

A necessidade do escalonamento de tarefas, respeitando a disponibilidade de recursos para as efectuar, é tão antiga quanto a actividade humana.

A verosimilhança da observação anterior facilmente decorre do entendimento que se tem do problema e que, essencialmente, se pode enunciar numa linguagem actual: dado um conjunto de tarefas, a serem efectuadas num sistema de processamento, encontrar a ordem (*sequence*) e os momentos de início e fim de processamento (*timetabling*) para cada tarefa e em cada processador, cumprindo um conjunto de condições e de modo a atingir o valor óptimo para um objectivo estabelecido.

Durante muitos séculos foi um problema com tratamento exclusivamente empírico e "avulso", marcando a década de 50 a passagem para uma abordagem sistémica e formal [Centeno, 1993]. Manteve-se, nesta última perspectiva, nos 15-20 anos seguintes um tema acentuadamente académico, até que a complexidade das aplicações pioneiras do *just-in-time production* (*JIT*), mostrou ser a única exequível, motivando o aprofundamento da base teórica e com um efeito indutor na aplicação destas técnicas a vastos sectores da economia.

A procura de soluções para o problema descrito em 2.4.2.2., que se fará neste trabalho, ilustra um pouco a parte final do último parágrafo.

Começar-se-á pela classificação dos escalonamentos seguindo [French, 1992], [Lawler et al, 1993] e [Brucker, 1995] (3.2. e 3.3.). Referem-se posteriormente propriedades das funções objectivo (3.4.) e analisa-se a existência de óptimo global (3.5.). Por último, descrevem-se métodos de determinação de escalonamentos para várias classes de problemas (3.6.).

3.2.Considerações preliminares

Um sistema de processamento (*machine environment*) é um conjunto de m máquinas M_i (i=1,2,..m) configurado para executar n tarefas (*jobs*) J_k (k=1,2,...,n). Cada tarefa consiste numa sucessão de operações, O_{ik}, a realizar por um subconjunto ou pela totalidade das m máquinas.

No âmbito deste trabalho, uma operação O_{ik} corresponde à "passagem" da tarefa k pela máquina i (*dedicated machines*) ou à passagem da tarefa k por uma, e só uma, das m máquinas (*parallel machines*). Excluem-se, assim, as situações de máquinas multifunções (*multi-purpose machines*) e multiprocessamento (*multiprocessor task sheduling systems*), [Brucker, 1995].

Um escalonamento (*schedule*) é a atribuição de um ou mais intervalos de tempo, numa ou mais máquinas a cada tarefa [Lawler et al, 1993]. É dito admissível se respeitar um conjunto de restrições gerais; umas genéricas, por serem comuns à grande maioria das situações estudadas, e também assumidas neste trabalho, nomeadamente: não serem atribuídos intervalos de tempo sobrepostos a tarefas distintas numa mesma máquina, a mesma tarefa não ter intervalos de tempo sobrepostos para ser processada em máquinas diferentes, cada tarefa não passar duas ou mais vezes pela mesma máquina, ou as

tarefas não poderem ficar parcialmente executadas; e outras mais exclusivas do tipo de problema. É óptimo se for um extremo global duma função representativa de um critério que se pretenda, correspondentemente, máximo ou mínimo.

Algumas restrições podem obrigar as tarefas a seguir uma ordem particular na cadeia que configura o sistema de processamento. Assim, se a sequência de execução nas máquinas for a mesma para todas as tarefas (todas cumprem a sequência: $M_{i\,(1)} \rightarrow M_{i\,(2)} \rightarrow \ldots \rightarrow M_{i\,(m)}$), designa-se o escalonamento como *flow-shop*. Se cada tarefa tiver que ser processada numa ordem pré-determinada, mas não necessariamente igual para todas as tarefas, é dito um *job-shop*. Um *open-shop* é um escalonamento que não tem que verificar nenhuma das restrições anteriores [Lawler et al, 1993][5].

O sistema de processamento (α), as características das tarefas (β) e o critério de optimalidade (γ), permitem definir [Lawler et al, 1993] e [Brucker, 1995], a classe $\alpha \mid \beta \mid \gamma$ do problema. Problemas que possuam a mesma estrutura $\alpha \mid \beta \mid \gamma$, são chamados instâncias da classe. A utilidade desta classificação é permitir reconhecer propriedades dos problemas (como elementos do grupo, ou inferi-las a partir de outras classes), que ajudem à sua solução.

[5] Em [French, 1992] é considerado outro tipo de escalonamento, *permutation schedule*, onde além da sequência de passagem nas máquinas ser igual para todas as tarefas (*flow-shop*), qualquer máquina processa as tarefas na mesma ordem ($J_{k(1)}, J_{k\,(2)}, \ldots, J_{k(n)}$).

3.3.Identificação da classe de escalonamento

3.3.1.Parâmetros de cada tarefa

A informação, suficiente para os propósitos deste estudo, usualmente associada a cada tarefa J_k (k=1,2,...,n) é a seguinte:

- Tempo de processamento de J_k na máquina M_i: p_{ik}. Quando o sistema de processamento é composto por uma só máquina pode resumir-se a: p_k;
- instante a partir do qual J_k está pronta para entrar no sistema (*ready time*): r_k;
- instante até ao qual J_k tem de ser concluída (*due time*): d_k;
- importância relativa (ponderador) de J_k no conjunto das tarefas (1,2,...,n): w_k.

Quando o número de tarefas, n, e os r_k são pré-determinados e constantes o problema diz-se estático por oposição a dinâmico. Igualmente quando as restantes variáveis são conhecidas e permanecem inalteradas ao longo do processo, o escalonamento chama-se determinístico em contraponto a estocástico [French, 1992].
No desenvolvimento que será fará no restante de 3.3., é pressuposto o caso estático e determinístico.

3.3.2.Sistema de processamento

O parâmetro α, tal como referido em 3.2., representa o sistema de processamento $\alpha=\alpha_1\alpha_2$, denotando neste contexto:

- $\alpha_1 = \circ$[6]: uma única máquina ($p_{1k} = p_k$);
- $\alpha_1 = P$: máquinas idênticas funcionando em paralelo ($p_{ik} = p_k$, $\forall$ i=1,2,...,m);
- $\alpha_1 = Q$: máquinas uniformes funcionando em paralelo (*uniform parallel machines*) ($p_{ik} = p_k/V_i$, onde V_i é a velocidade da máquina i);
- $\alpha_1 = R$: máquinas não relacionadas funcionando em paralelo (*unrelated parallel machines*) ($p_{ik} = p_k/V_{ki}$, onde V_{ki} é a velocidade da máquina i que depende da tarefa k);
- $\alpha_1 = J$: *job-shop*;
- $\alpha_1 = F$: *flow-shop*;
- $\alpha_1 = O$: *open-shop*.
- $\alpha_2 = m$: m representa o números de máquinas do sistema;
- $\alpha_2 = \circ$: O número de máquinas não é um valor pré-fixado, mas uma incógnita do problema.

É geralmente admitido que as máquinas podem, eventualmente, estar inactivas, que durante todo o período de escalonamento estão operacionais e que as restrições são pré-determinadas e mantidas.

[6] $\circ$ indica que o valor do parâmetro não é explicitado.

3.3.3.Características das tarefas

Em 3.3.1. explicitaram-se variáveis de cada tarefa. Os aspectos que se seguem são comuns ao conjunto $\{J_1, J_2, ..., J_n\}$ e determinam β, com $\beta=\beta_1\beta_2\beta_3\beta_4$, sendo:

- β_1 = pmtn: se uma operação O_{ik}, correspondente à passagem da tarefa J_k na máquina i, puder ser interrompida e recomeçada posteriormente (*preemption*);
- $\beta_1 = °$: caso contrário (uma operação O_{ik} uma vez começada terá que ser completada antes que outra possa começar na máquina i);
- β_2 = prec: se existirem tarefas que não possam ser realizadas antes de outras estarem concluídas;
- β_2 = tree: caso particular do anterior, em que as relações de precedência têm forma particular de árvore;
- $\beta_2 = °$: não são consideradas precedências.
- $\beta_3 = °$: todos os r_k têm o mesmo valor; (pode sem perda de generalidade considerar-se que: $r_k = 0, \forall k$);
- $\beta_3 = r_k$: existem tarefas com tempos distintos para entrada no sistema.
- $\beta_4 = p_{ik} = 1$: todas as tarefas têm, em qualquer máquina, o mesmo tempo de execução (correspondente a uma unidade de tempo);
- $\beta_4 = °$: caso contrário do anterior.

3.3.4.Critérios de optimalidade

O último termo, γ, na classificação de um escalonamento (na forma escolhida em 3.2.), representa um critério que permite medir o desempenho e, consequentemente, escolher o melhor escalonamento de entre todos os admissíveis.

Valores fulcrais no cálculo da *performance* de um escalonamento, são os momentos de conclusão (*completion times*), de cada tarefa J_k, simbolizados por C_k ($k = 1,...,n$) e dos quais se podem obter outros importantes indicadores para cada tarefa J_k como:

- tempo de permanência no sistema (*flow time*): $F_k = C_k - r_k$;
- atraso na conclusão (*tardiness*): $T_k = \max \{ C_k - d_k, 0 \}$;
- antecipação na conclusão (*earliness*): $E_k = \max \{ d_k - C_k, 0 \}$;
- diferença entre conclusão e *due time* (*lateness*): $L_k = C_k - d_k$;
- indicador de atraso (*unit penalty*): $U_k = 0$ se $C_k \leq d_k$; $U_k = 1$ se $C_k > d_k$.

Em [Centeno,1993], é salientado que as classes de problemas numa só máquina que têm sido mais estudadas são aquelas em que o critério de optimalidade é uma função de C_k ($k=1,2,...,n$).

Existem principalmente dois tipos de critérios:

$\gamma = \text{Max} \{ f_k(C_k); k=1,2,...,n \}$ e $\gamma = \sum_k f_k(C_k)$, sendo as funções denominadas, respectivamente, *bottleneck objectives* e aditivas (*sum objectives*), [Brucker, 1995].

Como exemplo referem-se médias (simples ou ponderadas por w_k), máximos e mínimos de C_k, F_k, T_k, L_k (k=1,2,...,n) e, merecedor de destaque para este estudo, $\gamma = \sum_k w_k U_k$.

Outra ilustração com apreciável aplicação está ligada aos problemas modelizados com base no conceito de *Just-in-Time.*

O *JIT* é definido como um método para suprimir na actividade das empresas investimentos improdutivos, sendo um exemplo comum a redução dos *stocks* nas várias fases do processamento [Centeno,1993].

Para este objectivo, é nuclear a minimização dos custos por antecipação ou atraso na conclusão das tarefas, formalizando-se o critério de desempenho do seguinte modo:

$Z = \sum_k a_k \times E_k + b_k \times T_k$, em que a_k e b_k são as penalidades, respectivamente, por antecipação e atraso na conclusão da tarefa J_k.

3.4.Propriedades das funções de desempenho

Existem classes de problemas de escalonamento cujas funções de desempenho partilham características com alcance suficiente para permitir uma abordagem idêntica, traduzida na possibilidade de empregar técnicas comuns de resolução.

A primeira, dessas propriedades, a observar é a de regularidade:

Uma função, R , dos tempos de conclusão das tarefas (C_1 , ... , C_n) é uma medida regular se $C'_k \leq C_k$, $\forall k$ $(k = 1,2, ..., n) \Rightarrow R(C'_1, ..., C'_n) \leq R(C_1, ..., C_n)$.

Significa então, que para um escalonamento, *S*, (com um critério que seja uma medida regular), cujos tempos de conclusão das tarefas sejam todos não superiores a um outro, S', terá, no mínimo, um desempenho tão bom quanto S'.

$\overline{C}$, C_{max}, $\overline{F}$, F_{max}, $\overline{L}$, L_{max}, $\overline{T}$, T_{max} e $\sum_k U_k$ são exemplos de funções regulares [French, 1992] (facilmente se prova que os respectivos mínimos também são critérios regulares). Inversamente, Z, $\overline{E}$ e E_{max} não são regulares [Centeno, 1993] e [French, 1992].

A maior parte da literatura referente a problemas de escalonamento dedica-se a classes com critério de desempenho regular, dada a maior facilidade de tratamento matemático.

Por exemplo, quando a função objectivo é regular e para sistemas de processamento com uma só máquina e *ready times* todos nulos, não se melhora o escalonamento forçando a máquina a inactividade, ou interrompendo tarefas [Morgan, 1993]. É fácil per-

ceber nestes casos a importância de regularidade, ao tornar estes problemas menos complexos, já que a solução se resume, exclusivamente, a encontrar a melhor entre todas as sequências de tarefas.

Em particular [Centeno, 1993], salienta o facto de em casos onde a função objectivo é não regular, não se pode fazer uso de teoremas de ordenação de tarefas, que facilitariam a sua resolução conjugados com algoritmos de numeração implícita. Igualmente de realçar é a referência em [Brucker, 1995], a desenvolver no Capítulo 6, de que problemas com uma máquina, tempos de processamento unitários e funções objectivo regulares do tipo *sum objectives*, puderem ser resolvidos como problemas de afectação.

Outro resultado, consequência da regularidade da função desempenho a merecer destaque, será exposto em 3.5..

Por outro lado, a aditividade é uma propriedade que se conjuga muito bem com as necessidades de métodos recursivos (como por exemplo a programação dinâmica).

Com as definições anteriores pode deduzir-se que se R é aditiva, é regular se e só se cada $f_k(C_k)$ é crescente [Morgan,1993]. (1)

Sendo $R=\sum_k w_k U_k = \sum_k w_k U_k(C_k, d_k)$, é um exercício fácil demonstrar que é aditiva e regular:

(i) R é aditiva:

Fazendo $f_k(C_k) = w_k U_k$, vem que $R=\sum_k w_k U_k = \sum_k f_k(C_k)$.

(ii) $f_k(C_k)$ é crescente para todo o k, com w_k e d_k pré-determinados e constantes:

Seja, como se disse, $U_k = \begin{cases} 0, \text{ se } C_k - d_k \leq 0; \\ 1, \text{ caso contrário} \end{cases}$ e $w_k \geq 0$:

$C'_k > C_k \Rightarrow U'_k \geq U_k \Rightarrow w_k U'_k \geq w_k U_k \Rightarrow f(C'_k) \geq f_k(C_k)$.

(iii) R é regular:

De (i) e (ii) vem que R é aditiva e $f_k(C_k)$ crescente $\forall k$, o que conjugado com (1), permite concluir que R é regular.■

3.5.O melhor escalonamento

Identificada a classe e para uma instância, o próximo passo é a obtenção do escalonamento que garanta o melhor valor para o critério escolhido. Surgem daqui dois tipos de problemas: a existência de óptimo e a forma de alcançá-lo.

Sendo os números de tarefas e de máquinas inteiros assumidamente fixados, uma forma de solução, primária, é a enumeração de todas as possibilidades de as sequenciar.

Ficaria, ainda assim, em aberto a decisão de se o valor do critério seria melhorado forçando as máquinas a tempos de inactividade ou, se possível, com interrupção de algumas tarefas.

A existência de óptimo, indiciada pela enumeração explícita de todas as hipóteses de sequenciamento, fica menos clara com as considerações introduzidas no último parágrafo.

Contudo, para um conjunto amplo de classes de escalonamento, é formalmente demonstrável que possuem óptimo e até uma metodologia para o atingir.

Define-se como *timetabling* semi-activo aquele em cujo escalonamento resultante, nenhuma operação pode antecipar-se sem alterar a sequência de processamento ou sem desrespeitar restrições.

Considere-se um sistema de processamento que obedeça às condições gerais enumeradas em 3.1. e 3.3.1., isto é:

- Uma tarefa não pode ser processada em simultâneo em duas máquinas;
- Uma tarefa é processada uma e uma só vez em cada máquina;
- Não são permitidas interrupções de tarefas;

e admita-se que o problema é estático, determinístico e γ uma medida regular.

Então o escalonamento óptimo é obtido a partir de *timetablings* semi-activos, sendo o número de escalonamentos gerados finito [French, 1992]. (**2**).

Seja escalonamento activo (*active scheduling*), um escalonamento em que nenhuma operação pode antecipar-se sem atrasar outra, ou sem violar restrições.

Conservando as condições anteriores, provam-se duas proposições com oportunidade para a existência do óptimo , [French, 1992]:

Que os escalonamentos activos são um subconjunto dos semi-activos (**3**) .

Que um escalonamento óptimo é activo (**4**).

Relacionando (**2**), (**3**) e (**4**) conclui-se, para um vasto conjunto de problemas de escalonamento, que existe extremo global para a função de desempenho.

A segunda questão, o método de atingir o óptimo, ultrapassa-se recorrendo ao algoritmo de Giffler e Thompson que, igualmente, nas condições de (**2**) permite gerar escalonamentos activos, [French, 1992] (**5**).

Invocando agora (4) e (5), deduz-se que, além da garantia de optimalidade, há uma forma geral e não exaustiva de resolver uma gama vasta de problemas de escalonamento[7].

Contudo, a teoria não é suficientemente desenvolvida para que exista uma formalização geral com aplicabilidade na procura do melhor escalonamento. As soluções vão ter que ser encontradas, para cada instância, em métodos ajustados à classe a que pertencem.

3.6.Métodos de resolução

Nesta secção far-se-á uma breve incursão por métodos de resolução para problemas de escalonamento, uns de aplicação generalizada, outros específicos de uma classe e desenvolvidos a partir de propriedades não partilháveis com outras classes. Alguns são exactos, enquanto outros não garantem a optimalidade (heurísticas).

Serão considerados métodos enumerativos, de pesquisa local, construtivos e encarar-se-á também a possibilidade de converter o problema do escalonamento num outro tipo, modelizando-o de forma a adequá-lo aos métodos desses tipos de problemas.

[7] Este método, com um grau de abrangência considerável só é aplicável, por dificuldades com o tempo de computação, a problemas com reduzido número de tarefas.

3.6.1.Enumerativos

Dividem-se em enumeração explícita (o caso trivial de todas as sequências possíveis) e de enumeração implícita, onde a pesquisa do óptimo é, de alguma forma, orientada evitando muitas sequências estéreis.

Dois métodos exactos, muito divulgados pela sua versatilidade, são a pesquisa em árvore (*branch and bound*) e a programação dinâmica.

Em [Lawler et al, 1993], referem-se trabalhos onde escalonamentos para as classes $1 \mid pmtn, r_k \mid \sum_k U_k$ e $1 \mid pmtn, r_k \mid \sum_k w_k U_k$, são determinados com técnicas de programação dinâmica e $1 \mid\mid \sum_k w_k U_k$ resolvido com recurso ao *branch and bound.*

O tratamento de problemas reais obriga contudo, quase sempre, a modificações onde se impõem regras empíricas para cancelamento de pesquisas, já que os tempos de computação apenas são não proibitivos para instâncias de pequena dimensão. São os casos do *beam search* e *approximate dynamic programming.* Não se garante o óptimo global, mas o ganho de eficácia computacional obtido torna-os exequíveis, mesmo para escalonamentos de dimensão considerável e com resultados satisfatórios.

A ideia nuclear é não averiguar todas as sequências que, potencialmente, possam conduzir a um valor melhor que o "corrente" para a função objectivo, mas limitar a pesquisa a um número fixo de ramificações, eleitas com base em regras adequadas ao problema e de cálculo fácil.

3.6.2.Técnicas de pesquisa local

As técnicas de pesquisa local, são também designadas por métodos heurísticos melhorativos ou de melhoramentos iterativos.

Neste tipo de procedimentos e na sua forma original procura-se melhorar uma solução inicial (vários métodos são possíveis para a obter), através da pesquisa a soluções próximas (a permuta de posições adjacentes é bastante comum), decorrendo o processo até que nenhuma deslocação beneficie o valor presente da função objectivo, significando estar-se num óptimo local.

Isto implica a definição de vizinhança de uma solução x, pertencente ao conjunto discreto de soluções X, de tal modo que para cada $x \in X$ é associado a um subconjunto $N(x) \subset X \wedge x \notin N(x)$, designado vizinhança de x. Pode então dizer-se que todos os vizinhos de x são as soluções obtidas a partir de x por movimentos considerados elementares.

Duas variantes podem ser utilizadas: "melhor vizinho " e "primeiro vizinho melhor". Na primeira o desvio é para a melhor solução da vizinhança (se existir), na segunda avança-se para a primeira solução vizinha que supere o valor da actual [Madureira et al, 1996].

Com a técnica descrita, os métodos de pesquisa local são muito intensivos, isto é, a melhor solução é procurada por pequenas deslocações a partir do ponto inicial, podendo criar-se uma notória situação penalizante quando o local de começo é muito afastado do óptimo.

Extensões, a esta forma "pura", foram desenvolvidas para possibilitar estratégias de diversificação, ou seja alargar o espaço de exploração de soluções admissíveis, tentan-

do escapar aos óptimos locais. Após análise da vizinhança, se o critério de finalização não foi ainda atingido, repete-se a busca num outro ponto x, escolhido por uma regra de selecção apropriada, mesmo que o desvio não seja proveitoso relativamente a uma solução já conseguida [Morgan, 1993].

Dois métodos são normalmente referidos: *Tabu search* e *simulated annealing*.

Com o primeiro, o novo valor escolhido é da melhor valorização da função objectivo entre todas as vizinhas, N(x) ou entre as de um subconjunto $N'(x) \subset N(x)$. Para prevenir eventuais retornos, durante um número arbitrado de iterações, a "locais já visitados", é elaborada uma lista (*tabu list*), com movimentos efectuados (todos ou um número fixado), que é consultada para os evitar.

O processo termina com a satisfação de um critério que pode ser o tempo de cálculo ou o número de movimentos [Morgan, 1993], [Madureira et al, 1996].

O *simulated annealing* é um método inspirado no processo físico de cristalização por arrefecimento estudado na termodinâmica e na metalurgia.

Ao contrário do anterior, as deslocações são escolhidas de forma não determinística. São calculados os valores da função objectivo para cada hipótese de movimento e a probabilidade de optar pelo ponto g é: $p_g \propto e^{-\left[\frac{K}{K_0} \times \left(D_{max} - D_g\right)\right]}$, onde D_g é o valor da função objectivo caminhando para g e D_{max} o máximo de entre todas as alternativas. K_0 é um factor de normalização que faz com que a soma das probabilidades de uma vizinhança seja unitária e $\frac{K}{K_0}$ a "temperatura normalizada".

A "temperatura" vai descendo ao longo do processo, que termina quando se atinge um estado pré-determinado de arrefecimento [Morgan, 1993].

3.6.3.Métodos construtivos

Ao contrário dos métodos melhorativos, os construtivos partem de uma solução inicial vazia e vão em cada iteração dos algoritmos construindo uma solução parcial, atingindo no final do cálculo uma solução completa para o problema.

Existem poucas classes de problemas que tenham algoritmos construtivos optimais, especialmente sistemas de processamento com mais de uma máquina.

Em [Lawler et al, 1993], são inventariadas classes de problemas para as quais foram desenvolvidos algoritmos construtivos, todas com critérios de desempenho regulares.

Relevante para este trabalho, como se verá adiante, é o algoritmo de Moore-Hodgson, para a classe $1 \mid\mid \sum_k U_k$, constituído nos passos seguintes [French,1992]:

Algoritmo de Moore-Hodgson

1.Ordenar todas as tarefas por ordem crescente dos *due times*, formando a sequência: $(J_{u(1)}, J_{u(2)}, \ldots, J_{u(n)})$, onde $d_{u(k)} \leq d_{u(k+1)}$, k=1,2,..., n-1.

2.Detectar a primeira tarefa em atraso $J_{u(k)}$ na sequência anterior. Se nenhuma for encontrada, então ir para 4.

3.Seleccionar a tarefa de $(J_{u(1)}, J_{u(2)}, \ldots, J_{u(k)})$, com o maior tempo de processamento e excluí-la. Voltar a 2 com um número de tarefas reduzido de uma unidade.

4.A sequência resultante, a que se juntam a partir da última tarefa concluída em *due time*, as tarefas rejeitadas em 3 (e adicionadas em qualquer ordem), é o escalonamento óptimo.

É feito notar que a classe $1 \mid\mid \sum_k w_k U_k$, tem uma solução facilmente derivável do caso anterior, quando $p_k < p_{kl} \Rightarrow w_k \geq w_{kl}$[8].

[Kise, Ibari e Mine, 1978], desenvolveram um método aplicável a $1 \mid r_k \mid \sum_k U_k$ se $r_{kl} < r_k \Rightarrow d_{kl} \leq d_k$.

Para as classes $1 \mid r_k, p_k = 1 \mid \sum_k f_k(C_k)$, sendo $f_k(C_k)$ regular, em [Brucker, 1995], é mostrado como transformar estes problemas de escalonamento em problemas de afectação e demonstrado que se atinge o valor óptimo para a função objectivo.

Um método heurístico construtivo para funções objectivo regulares, de características generalistas, é exposto em [French, 1992].

Defina-se escalonamento sem atraso (*non-delay scheduling*), aquele onde nenhuma máquina é deixada inactiva se puder processar alguma tarefa.

Esta categoria de escalonamentos forma uma subclasse dos escalonamentos activos, em número bastante inferior. Um escalonamento óptimo não é necessariamente *non-delay*, mas, empiricamente, admite-se que o melhor destes escalonamentos não está muito afastado do óptimo[9].

Non-delay schedulings podem ser gerados, continuando a respeitar as condições de (2), com modificações ligeiras do algoritmo de Giffler e Thompson, [French, 1992].

---

[8] Em [Morgan, 1993], é descrito um algoritmo para $1 \mid\mid \sum_k w_k U_k$ quando $p_k = 1$, $\forall k$ que, obviamente, é um caso que verifica a relação $p_{kl} < p_k \Rightarrow w_{kl} \geq w_k$

[9] Pode ainda tentar-se melhorar uma solução obtida desta forma, usando-a como ponto inicial de um método de pesquisa local.

CAPÍTULO 4

GERAÇÃO DE SOLUÇÕES PARA O PROBLEMA DA ALIMENTAÇÃO DAS DIVISORAS COM BASE NUM MODELO DE ESCALONAMENTO DE TAREFAS

"Consideremos uma vaca esférica..."[10]

Admita-se resolvido o problema enunciado em 2.4.2.1., isto é, estão determinados o número de saídas dos *OCR*, m, e o agrupamento de destinos em cada saída. Consequentemente estão identificadas, para cada saída dos *OCR*, as horas limite de passagem nas máquinas de divisão, a quantidade de cartas e a prioridade ficando, assim, definidos os parâmetros[11] que permitem configurar o problema 2.4.2.2..

É intenção nesta fase considerar o problema como um caso particular de escalonamento e conceber uma forma de alimentação das máquinas de divisão, isto é, organizar a passagem das cartas (das várias saídas dos *OCR)*, de modo a minimizar o número de não divididas dentro das horas limite, respeitando prioridades, mas com tempos de esvaziamento das divisoras nulos.

Começar-se-á por situar o problema na óptica das noções introduzidas em 3. (4.1.). Em seguida (4.2.), discutir-se-à um modelo de funcionamento para o sistema de divisão mecanizada, com uma só máquina e sem conflituar com a prática operacional, que

[10] Em [Ian Stewart,1996], é contado um episódio onde "Um agricultor, preocupado com o facto de a sua manada não estar a produzir leite suficiente, contratou a Universidade local para investigar. Foi formada uma equipa interdisciplinar, chefiada por um matemático aplicado. Passado um mês, esta equipa tinha preparado um relatório com 100 páginas. Desejoso de descobrir o que tinha comprado com o seu precioso dinheiro, o agricultor sentou-se e começou a ler - apenas para se lhe deparar a frase de abertura: «Consideremos uma vaca esférica...»"

[11] A hora limite de passagem é, recorda-se, a do destino com menor hora limite pertencente ao agrupamento; a quantidade de cartas em cada saída o acumulado de todos os destinos dessa saída; a prioridade de cada saída o máximo de entre as prioridades dos destinos dessa saída.

conduzirá à escolha da classe do escalonamento entre as alternativas enumeradas em 4.1. Na secção imediata (4.3.), propor-se-á um método para determinação do escalonamento nas condições adoptadas e discute-se (4.4), a validade do método apresentado. Posteriormente, estender-se-ão estes resultados para duas máquinas, mantendo todas as restantes hipóteses do modelo (4.5.). Finalmente, encarar-se-á a possibilidade de prioridades diferenciadas (4.6.).

4.1.Enquadramento

A identificação da classe do problema 2.4.2.2., na continuidade do exposto em 3., dependerá evidentemente da modelização que se eleger.

Das possíveis alternativas que se irão expor, escolher-se-à uma, que além de satisfazer requisitos de aderência à realidade, se associe com um método de resolução fácil de pôr em prática e que proporcione resultados satisfatórios.

Para o problema que se pretende tratar (continuando com o critério de [French, 1992]), embora a quantidade de cartas que chega diariamente a cada CTC não seja fixa, o escalonamento é transformado em estático, já que se irá trabalhar, como se disse, com valores médios.

No caso de não serem considerados tempos de esvaziamento das divisoras (uma das possibilidades apontadas em 2.4.2.2.), o escalonamento será determinístico. Admitindo a possibilidade inversa, poderá conceptualizar-se como sendo um problema em que o

tempo de processamento é dependente da sequência das tarefas[12]; hipótese que não será assumida agora, mas que se retomará no Capítulo 5.

4.1.1.Sistema de processamento

No caso presente existem, como se disse, dois CTC (Coimbra e Porto), com uma só máquina de divisão e o CTC de Lisboa com duas em paralelo. São, assim, possíveis dois sistemas de processamento:

- uma máquina: $\alpha_1 = °$, $\alpha_2 = 1$ e $\alpha=1$;

- duas máquinas em paralelo: $\alpha_1 =P$, $\alpha_2 = 2$ e $\alpha=P2$.

As considerações sobre inactividade, operacionalidade e manutenção das restrições feitas em 3.3.2., são apropriadas a esta situação.

4.1.2.Características das tarefas

- Cada tarefa pode entender-se como a triagem de uma carta, ou um lote de cartas sem permitir interromper a operação, o que se traduz por $\beta_1 = °$;

ou

- Cada tarefa a triagem de um lote de cartas que não constitua uma unidade, podendo então supor-se *preemption* vindo $\beta_1 =$ pmtn.

[12] Tomando como tempo de processamento a soma do tempos de divisão e esvaziamento.

Como não fazem sentido regras de precedência: $\beta_2 = °$.

Como as cartas chegam em momentos diferenciados: $\beta_3 = r_k$.

Mais alternativas são possíveis para β_4:

- Cada tarefa a triagem de uma carta ou de um lote com um número fixo de cartas sem permitir interromper a operação, sendo a unidade temporal o tempo de divisão do que se tome como tarefa: $\beta_4 = p_k = 1$, obtendo-se $\beta = r_k, p_k = 1$;

- Cada tarefa a triagem de um lote com um número variável de cartas sem permitir interromper a operação: $\beta_4 = °$, vindo $\beta = r_k$;

- Cada tarefa a triagem de um lote com um número fixo de cartas, permitindo interromper a operação e a unidade de tempo o tempo de divisão de um lote: $\beta_4 = p_k = 1$, resultando $\beta = pmtn, r_k, p_k = 1$;
- Cada tarefa a triagem de um lote com número variável de cartas e permitindo interromper a operação: $\beta_4 = °$, resultando $\beta = pmtn, r_k$.

4.1.3.Critério de optimalidade

O critério a seguir neste escalonamento será, obviamente, minimizar $\gamma = \sum_k w_k U_k$, representando w_k a ponderação devida às prioridades. Se forem todas iguais: $w_k = 1$ e $\gamma = \sum_k U_k = n_T$.

4.1.4.Classe do escalonamento

Como conclusão, podem agora estabelecer-se as possíveis classes em função do modelo de abordagem deste problema:

- 1 máquina:

- tarefa ≡ triagem de uma carta ou triagem de um lote com um número fixo de cartas, sem permitir interrupção: $1 \mid r_k, p_k=1 \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número variável de cartas sem permitir interrupção: $1 \mid r_k \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número fixo de cartas permitindo interrupção: $1 \mid pmtn, r_k, p_k=1 \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número variável de cartas permitindo interrupção: $1 \mid pmtn, r_k \mid \sum_k w_k U_k$;

- 2 máquinas:

- tarefa ≡ triagem de uma carta ou triagem de um lote com um número fixo de cartas, sem permitir interrupção: $P2 \mid r_k, p_k=1 \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número variável de cartas sem permitir interrupção: $P2 \mid r_k \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número fixo de cartas permitindo interrupção:

$P2 \mid pmtn, r_k, p_k=1 \mid \sum_k w_k U_k$;

- tarefa ≡ triagem de um lote com um número variável de cartas permitindo interrupção: $P2 \mid pmtn, r_k \mid \sum_k w_k U_k$;

4.2.Modelização

Das opções de modelização anteriores, as classes que tem referência na literatura consultada, como tendo um algoritmo desenvolvido e exacto, são, como se viu:

- $1 \mid pmtn, r_k \mid \sum_k w_k U_k$, [Lawler et al, 1993];
- Um caso particular de $1 \mid r_k \mid \sum_k U_k$, [Kise, Ibari e Mine,1978][13];
- $1 \mid r_k, p_k=1 \mid \sum_k w_k U_k$, [Brucker, 1995].

O algoritmo para a primeira classe tem tempo de computação da ordem da quinta potência do número de tarefas quando $w_k=1$, ou proporcional a $n^3(\sum_k w_k)^2$, quando $\exists w_{k1} \neq w_k \wedge w_k, w_{k1} \in N \ \forall k,k1$. No caso mais simples de $w_k=1, \forall k$, como diariamente o CTC de Lisboa recebe cerca de 1.8 milhões de cartas, mesmo fazendo lotes de 15000

[13] Evidentemente que daqui se retiram soluções para os casos particulares com $p_k=1$.

cartas correspondendo a cerca de 90 tarefas recebidas antes da hora de corte, chegar-se-iam a valores de tempos de computação absurdos[14].

A modelização do problema como sendo da segunda classe não é adequado por não se verificar que $r_{k1} < r_k \Rightarrow d_{k1} \leq d_k$.

A última alternativa, tem tempo de computação aceitável (é calculada em tempo proporcional à terceira potência do número de tarefas). Contudo, tomando lotes de 10000 cartas é necessário considerar cerca de 140 tarefas[15], tornou-se difícil conseguir *software* que "corresse" em microcomputador para problemas destas dimensões.

Outro inconveniente, de natureza operacional, que se explicará adiante, é o não favorecimento por este método de escalonamentos em que, sempre que possível, os lotes duma mesma saída se sucedam na alimentação das divisoras.

Concluindo-se que estas hipóteses, com métodos de resolução comprovadamente optimais, não são inteiramente satisfatórias, procurou-se, dentro do conjunto considerado em 4.1.4., outra alternativa. Especificam-se no que se segue, justificando as escolhas, o sistema de processamento, características das tarefas, critério de optimalidade e assumpções relacionadas com o método que se irá propor.

- Sistema de processamento

- uma máquina ≡ uma divisora;

- regras de funcionamento do sistema:

As condições (compatíveis com a realidade), em que o sistema de processamento deve funcionar e que o procedimento que se irá desenvolver adopta, são as seguintes:

[14] Para um tempo de cálculo de um milésimo de segundo para cada tarefa, 90 tarefas precisariam de cerca de 69 dias. Existem também, como se justificará, desvantagens na admissão das tarefas puderem serem interrompidas.

[15] Recebidas anteriormente aos respectivos *due times*.

. O tempo total de operação do sistema é dividido em L períodos de duração Δ_t.

A aleatoriedade intrínseca à actividade postal não permite conhecer o instante da chegada de cada carta, mas apenas médias de quantidades recebidas por destinos em intervalos de tempo com amplitude suficiente para garantir estabilidade aos valores estimados. Esta condicionante será conformadora de uma abordagem para o problema.

No estudo mais recente, foram determinados fluxos médios de cartas chegadas aos CTC por destinos e hora-a-hora, o que levará a que todas as informações quantitativas disponíveis sejam referentes a cada período de uma hora. Decorre daqui o sentido da divisão do tempo de funcionamento, e mais concretamente fazer Δ_t= 1 hora.

Sabem-se, assim, as quantidades de cartas para dividir recebidas até ao início do primeiro período e em cada período i ($1 \leq i < L$[16]), e por saída dos *OCR*. Clarificando: no intervalo $[\Delta_t \times i\ ,\ \Delta_t \times (i+1)[$, chegam Q_i^1 cartas da primeira saída, Q_i^2 cartas da segunda, Q_i^m cartas da m-ésima.

Representando as cartas da saída j que existem no começo do primeiro período por Q_0^j, cada saída j dos *OCR* origina, portanto, $Q_0^j + Q_1^j + Q_2^j + ... + Q_{L-1}^j$ cartas.

. A contagem do tempo é feita com referência ao início do primeiro período;

• Tarefas

- tarefa ≡ triagem de um lote de cartas em número fixo (proveniente de qualquer saída dos *OCR*) sem permitir interrupção.

A inevitabilidade da pré-divisão, faz com que os parâmetros de cada carta se confundam com as do agrupamento a que pertence.

Conceber o escalonamento para a divisão carta-a-carta, fundado nos *ready times* e *due dates* reais, obrigaria à identificação prévia e precisa de cada carta à saída dos *OCR*, significando que se estava num caso extremo onde, após a indexação, estariam já separadas por destinos finais.

Explicada a escolha de cada tarefa corresponder a um lote, existem duas ordens de razões para que sejam unos e de tamanho fixo:

. O método de resolução (a que adiante se fará referência);

. Ser um obstáculo do ponto de vista operacional a interrupção da divisão, excepto se se admitisse a existência de um limite mínimo de processamento, para evitar a mudança frequente de programas nas divisoras[17].

Representando por D[18] a dimensão de cada lote, o número total de tarefas a processar virá: $\sum_j \left\lfloor \frac{\sum_{i=0}^{L-1} Q_i^j}{D} \right\rfloor$. O resto da divisão, para cada saída j, serão cartas excluídas do processo mas, atribuindo a D um valor "razoável", não terá significado.

- instante até ao qual a tarefa tem de estar concluída ≡ hora de corte do destino com menor hora de corte dentro do agrupamento a que corresponde o lote;

Para garantir que todas as cartas de um lote são divididas até *due date*, é suficiente que o lote seja processado até ao momento coincidente com o destino de menor hora de corte dentro do grupo.

---

[16] As cartas recebidas em L já não são consideradas.

[17] Justifica-se também desta forma que modelizar como um problema de afectação para determinar o escalonamento, poderia traduzir-se em comutações indesejáveis de programas nas máquinas de divisão, por não ser beneficiada a formação de lotes contíguos da mesma saída dos *OCR*.

[18] Existe um modo expedito de contar cartas, pois o transporte dentro de cada CTC é feito em tabuleiros com capacidades várias mas múltiplos do milhar de cartas (± poucas unidades).

- instante para uma tarefa entrar no sistema

As cartas processáveis no período i, são aquelas que chegam até ao limite inferior de i.

Como já salientado, apenas existem dados quantitativos para cada período de uma hora. Por esta razão, um lote só é admitido como disponível no início do período seguinte aquele em que foi recebido.

O que se entende por *ready time,* no contexto do modelo que se apresenta, será clarificado quando for descrito o método proposto para determinação do escalonamento.

- tempo de processamento de uma tarefa $\Leftrightarrow$ tempo de divisão de um lote (p = 1 unidade de tempo);
- ponderador de cada tarefa $\equiv$ prioridade de cada lote = 1.
- não podem ser interrompidas;
- não existem precedências;

• Critério de optimalidade

- minimizar lotes não divididos dentro das horas de corte $\equiv \sum_{k} U_k = n_T$.

O objectivo ideal seria minimizar o número de cartas em atraso e não o número de lotes formados com os agrupamentos correspondentes a saídas dos *OCR*. Minimizando o número de lotes, dada a variabilidade de *due times* dentro de cada grupo, poderá inviabilizar-se o processamento de algumas cartas antes das horas de corte. Com as limitações já descritas julgou-se, contudo, ser o único critério possível.

- Classe do problema

O modelo descrito até agora parece apontar para a classe: $1 \mid r_k, p_k=1 \mid \sum_k U_k$.

O método que se proporá, e que se descreve em 4.3. supõe em cada período $r_k = 0$, $\forall_k$, transformando o problema em cada período i na classe: $1 \mid p_k=1 \mid \sum_k U_k$.

4.3.Algoritmo de resolução

Definido o problema, importa agora tratar da sua resolução.

O método que se propõe para determinar o escalonamento assenta no algoritmo de Moore-Hodgson , exposto em 3.6., o que explica a hipótese de não *preemption.*

Mesmo no caso mais simples que aqui se analisa, para que o algoritmo de Moore-Hodgson seja adequado, restaria o obstáculo de os *ready times* não serem todos nulos, isto é, os lotes não estarem todos no mesmo instante prontos para serem processados.

A ideia base que aqui se sugere é, como já se adiantou, ultrapassar esta dificuldade aplicando repetidamente o algoritmo de Moore-Hodgson a cada um dos L períodos em que se divide o tempo total de funcionamento do sistema, considerando que em cada período i, todas as tarefas chegadas e não processadas até ao início de i, têm *ready times* nulos.

No início de cada período i, o número de tarefas de cada saída j, é igual ao quociente da divisão inteira das cartas chegadas para a saída j e ainda não processadas, por D. O resto juntar-se-á às cartas do período seguinte, como se tivessem chegado em

$[i\times\Delta_t, (i+1)\times\Delta_t[$[19]. Os lotes que não se conseguiram processar no período i, adicionam-se igualmente às cartas chegadas em $[i\times\Delta_t, (i+1)\times\Delta_t[$[20].

A decisão de não trabalhar com lotes de comprimento variável potencialmente incrementará o número de cartas divididas com atraso[21] , pois cartas que poderiam ser tratadas no período i são deslocadas imediatamente para o período i+1. Apesar disso, se o tamanho dos lotes for "pequeno" , dado o volume total de correspondências, será desprezável.

A dimensão dos lotes de cartas, D, resultará de uma avaliação entre as cartas que poderão deixar de ser divididas dentro das horas de corte, por D ser elevado, e as mudanças de programa de divisão que poderão resultar, se D for diminuto.

Vantagens na utilização de lotes de tamanho fixo também existem e reflectem-se na simplificação e eficiência do algoritmo:

- A ordenação crescente das tarefas em cada período por *due time* e o facto de o tempo de processamento ser igual para todas as tarefas, faz com que assim que se

---

[19] Note-se que o total de lotes em todo o período de funcionamento do sistema mantém-se em:

$$\sum_j \left\lfloor \frac{\sum_{i=0}^{L-1} Q_i^j}{D} \right\rfloor$$

[20] Por exemplo, a totalidade dos lotes gerados até ao começo do primeiro período é:

$1,\ldots, J_0^1, J_0^1+1,\ldots, J_0^1+J_0^2,\ldots,1+\sum_{j=1}^{m-1} J_0^j,\ldots,\sum_{j=1}^{m} J_0^j$, onde:

$$J_0^1 = \left\lfloor \frac{Q_0^1}{D} \right\rfloor,\ J_0^2 = \left\lfloor \frac{Q_0^2}{D} \right\rfloor,\ \ldots,\ J_0^m = \left\lfloor \frac{Q_0^m}{D} \right\rfloor.$$

As cartas de cada saída j do *OCR* insuficientes para preencher um lote no período i=1: $Q_0^j - J_0^j \cdot D$, consideram-se como se tivessem sido recebidas no intervalo seguinte, isto é somam-se a Q_1^j.

[21] Já prejudicado pela diferenças de *due times* no mesmo lote

detecte a impossibilidade de processar um lote no período i, por ultrapassagem do tempo limite, esse lote seja excluído para ser dividido em i.

- Com lotes de tamanho constante, a sua dimensão poderá ser definida de modo aos intervalos de tempo Δ_t serem múltiplos de p^{22}. Deriva daqui, que todas as tarefas iniciadas em i são concluídas em i.

Três modificação que se introduziram, relativamente à descrição do algoritmo em 3.6., são:

- A verificação de se alguma saída do *OCR* tem no início do período i *due time* inferior, ou igual a $\Delta_t \times i$. Se tal ocorrer, e por razões de velocidade de computação, altera-se o *due time* para $\Delta_t \times (L+1)$.

  Evita-se deste modo estar em todos os períodos a rejeitar os lotes dessa saída, que só poderão voltar a ser processados se todas as outras saídas estiverem também para além dos *due times*.

- A ordenação dos lotes, já recebidos até ao início do período i e ainda não processados, por *due time* e saída.

  Se existirem várias saídas com o mesmo *due time,* sem violar o algoritmo de Moore-Hodgson, os lotes de uma saída são processados sem intercalação de lotes dessas outras saídas, não promovendo as mudanças de programa nas divisoras.

- O cálculo, em cada período i, da variável *Due time* = $\text{Min}\{$ *due time*, $\Delta_t \times (i+1)\}$ para determinar até quando um lote pode ser triado, e sem atraso, no intervalo de tempo em que decorre o período i.

  Note-se que os lotes ordenados por *due time* e saída ficam igualmente ordenados por *Due time* e saída.

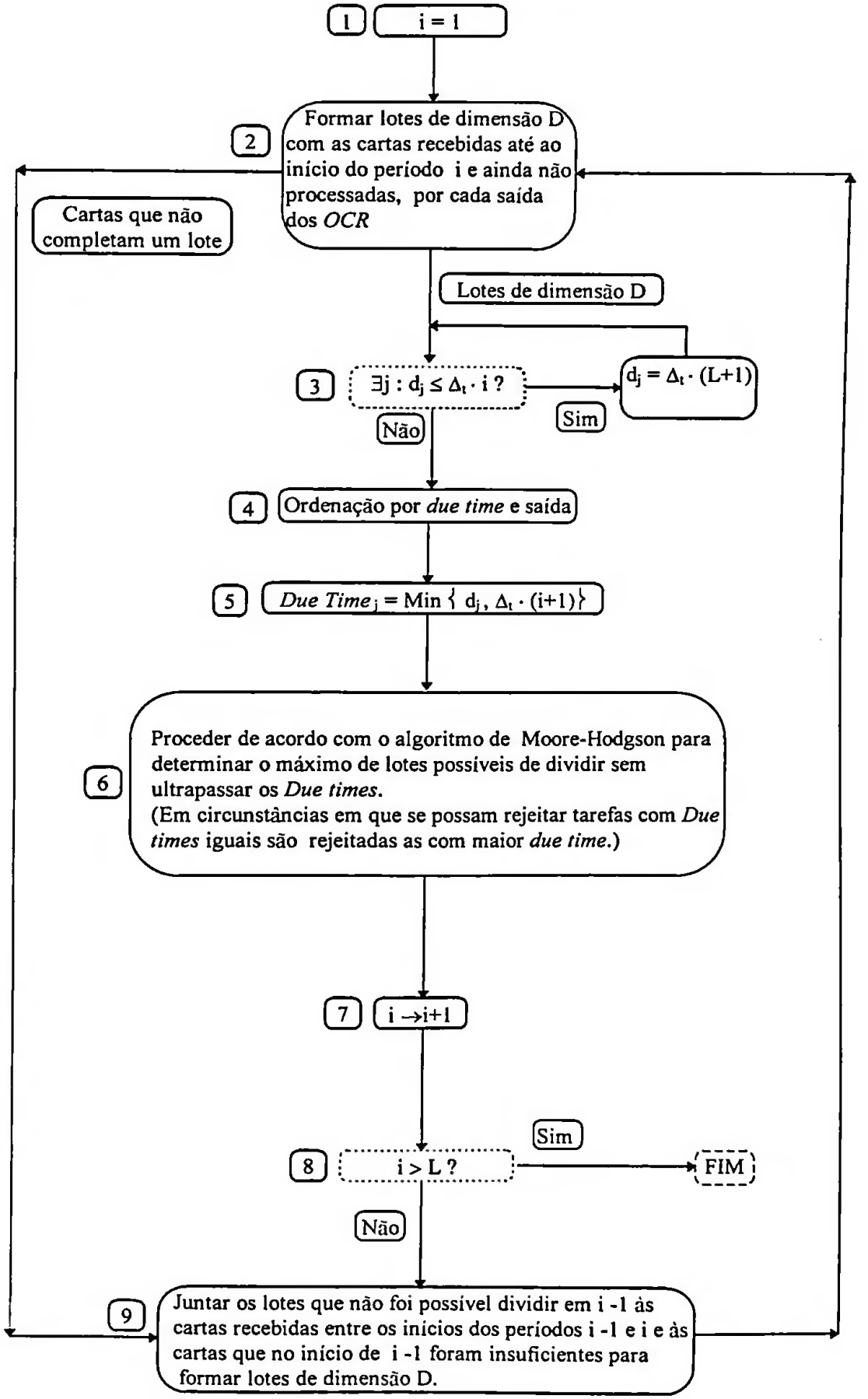
O tempo de execução depende, essencialmente, dos processos de ordenação e de operações de comparação, para cada lote, de *due times* (e *Due times*) com o tempo decorrido desde o começo do funcionamento do sistema. Como estas comparações têm tempo de execução proporcional ao número de lotes e usando como técnica de ordenação o "*bubble sort*" cujo tempo de cálculo é função do quadrado do número de lotes, resulta um tempo de computação polinomial, de ordem dois, relativamente ao número de tarefas.

O detalhe de construção do escalonamento *S*, é descrito no fluxograma seguinte:

[22] Cada 1000 cartas têm um tempo de processamento de 2 minutos nas máquinas dos CTC de Lisboa e Porto e 3 minutos na de Coimbra.

Figura 2

Algoritmo proposto



4.4.Validade do algoritmo

Proposição P1: O método proposto gera um escalonamento, S, que maximiza o número de lotes divididos dentro da respectiva hora de corte para todo o período de funcionamento da máquina.

Demonstração:

1.Prova da existência de um escalonamento óptimo.

Recordando 3.5., nas condições do problema descrito em 4.2. existe um escalonamento óptimo pertencente ao conjunto do *timetablings* semi-activos.

O facto do melhor escalonamento ser semi-activo permite concluir, também, que o escalonamento óptimo só admite tempos de inactividade da máquina de divisão, em qualquer intervalo, se não houver lotes disponíveis para dividir nesse intervalo.

2.Seja S^* um escalonamento óptimo para o problema referido, onde $n_H(S^*)$ é o número máximo de lotes que é possível dividir dentro das respectivas horas de corte ao longo de todo o período de funcionamento da máquina.

Pretende-se provar que o escalonamento obtido com o método proposto, S, é tal que $n_H(S) = n_H(S^*)$.

Se $S \equiv S^*$ o resultado é trivial.

Se $S \neq S^*$ existe pelo menos um lote que em S ocupa uma posição diferente daquela que ocupa em S^*.

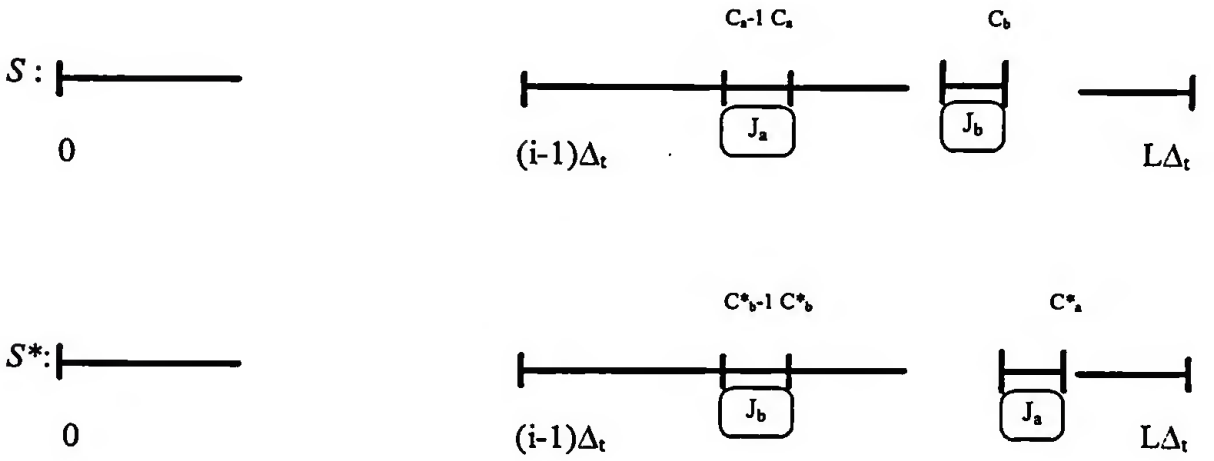
Seja J_a o primeiro lote nessas condições.

Sejam d_a a sua hora de corte, C_a o seu momento de conclusão em S e C_a^* o seu momento de conclusão em S^*.

Como S^* (por hipótese de óptimo) e S (por construção), são semi-activos, existe um outro lote, J_b, que em S^* ocupa a posição de J_a em S, isto é, que em S^* é dividido em $[C_a\text{-}1, C_a]$ ($p_a=p_b$= uma unidade de tempo). Esquematicamente:

Figura 3

Posição dos lotes J_a e J_b nos escalonamentos S e S^*



Se no escalonamento S^* o lote J_b é dividido em $[C_a\text{-}1, C_a]$ é porque está disponível para a divisão, isto é, $r_b \leq (i\text{-}1)\,\Delta_t$.

Dois casos se podem dar:

2.1. $C_a \leq d_a$

Em S o lote J_a está a ser processado dentro da sua hora de corte. Como S resulta da aplicação do algoritmo de Moore-Hodgson no intervalo $[(i\text{-}1)\Delta_t, i\Delta_t[$, novamente duas situações podem ocorrer:

2.1.1.$d_b < d_a$

Nesta condição J_b já ultrapassou a sua hora de corte, isto é, $C_b > d_b$. Trocando a posição de J_a com J_b em S^* resulta um escalonamento S^1 onde $n_H(S^1) \geq n_H(S^*)$, que por S^* ser óptimo obriga a que $n_H(S^1) = n_H(S^*)$.

2.1.2. $d_b \geq d_a$

Considere-se mais uma vez o escalonamento S^1 obtido pela troca de posições entre J_a e J_b em S^*.

Em S^1 os momentos de conclusão serão $C^1_a = C_a \leq d_a$ e $C^1_b = C^*_a$. Como $d_b \geq d_a$ vem $n_H(S^1) \geq n_H(S^*) \Rightarrow n_H(S^1) = n_H(S^*)$, por S^* ser, por hipótese, óptimo.

Em S^1 há mais um lote do que em S^*, J_a, cuja posição coincide com a que ocupa em S.

2.2.$C_a > d_a$

Neste caso sabe-se que $d_b < C_a = C^*_b$ pois, como $r_b \leq (i-1)\,\Delta_t$, se fosse $C_a \leq d_b$ na construção de S o lote J_a não teria sido posicionado em $[C_a-1, C_a]$.

De novo trocando as posições de J_a e J_b em S^* obtém-se um escalonamento S^1 com $n_H(S^1) = n_H(S^*)$ e, tal como em 2.1. o escalonamento S^1 tem mais um lote, J_a, cuja posição coincide com a que ocupa em S.

Se $S^1 \equiv S$ então $n_H(S) = n_H(S^1) = n_H(S^*)$ e está provado o resultado. Caso contrário pode repetir-se para S^1 e S o raciocínio feito para S^* e S.

Repetindo o raciocínio tantas vezes quantas as necessárias (no máximo total de lotes em S^* (n)- número de lotes processados até C_a), resulta uma sequência de escalonamentos $S^1, S^2, \ldots, S^{p-1}, S^p$, tal que:

$n_H(S^*) = n_H(S^1) = n_H(S^2) = \ldots = n_H(S^{p-1}) = n_H(S^p)$, em que $S^p \equiv S$ e portanto $n_H(S^*) = n_H(S)$, ficando provada a proposição. ∎

4.5.O problema com duas máquinas

Retome-se agora o escalonamento para duas máquinas em paralelo.

No passo 4 do procedimento 4.3., recorda-se, impõe-se que o critério de ordenação seja composto de uma chave principal (*due time*) e outra secundária (saída dos *OCR*). Esta operação, sem violar o algoritmo de Moore-Hodgson, promove a formação de lotes contíguos da mesma saída, que são introduzidos sucessivamente durante um intervalo $[i\cdot\Delta_t, (i+1)\cdot\Delta_t[$ até ao esgotamento, isto é, que todos os lotes da saída j dos *OCR*, a dividir no intervalo i, sejam divididos sem lotes de outra saída q ($q \neq j$), com *due times iguais*, a interporem-se[23].

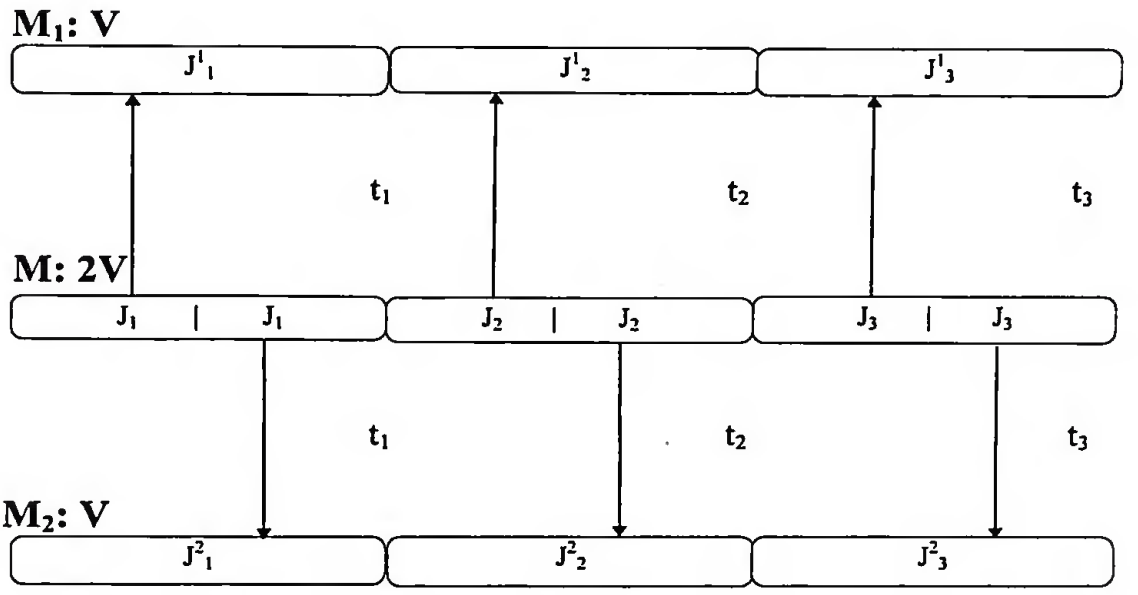
O modelo e o algoritmo desenvolvidos anteriormente para uma máquina podem ser facilmente adaptados para esta situação de duas máquinas, resolvendo um problema para uma máquina fictícia com velocidade dupla da velocidade das máquinas reais, isto é, na qual os tempos de processamento dos lotes são 50% dos reais. Uma vez obtida a sequência para a máquina fictícia, os lotes são divididos ao meio e cada metade afecta a uma das máquinas reais, mantendo para elas a sequenciação obtida para a máquina fictícia como mostrado na próxima figura:

[23] Não teria sentido prático, como se disse, a mudança frequente de programa na máquina de divisão.

Figura 4

Escalonamento para duas máquinas a velocidade V

a partir de um escalonamento para uma máquina a velocidade 2V



Qualquer escalonamento construído, segundo 4.3., para uma máquina que tenha uma velocidade de processamento 2V pode ser, deste modo, transposto para duas máquinas operando em paralelo com velocidade V.

O fraccionamento dos lotes em duas partes iguais constitui, do ponto de vista teórico, uma alteração importante ao modelo pois a hipótese de que uma tarefa quando iniciada tem que ser concluída sem interrupção (*no preemption*), é em muitos casos fundamental no desenvolvimento da teoria.

No caso em estudo a divisão dos lotes ao meio não introduz dificuldades operacionais relevantes e permite, como se vai demonstrar, obter para duas máquinas resultados equivalentes aos já obtidos para uma máquina.

Lema L1:

Seja $S = [\ J_{k_1}, J_{k_2}, \cdots, J_{k_n}\ ]$ um escalonamento do conjunto de n tarefas $\{J_1, J_2, \cdots, J_n\}$, com tempos de processamento p_k $(1 \leq k \leq n)$, numa máquina com velocidade 2V.

Considere-se cada lote dividido em duas partes iguais[24]: $J_{k_i} = (J^1_{k_i}, J^2_{k_i})$, e formem-se dois escalonamentos mantendo a ordem em S: $S^1 = [\ J^1_{k_1}, J^1_{k_2}, \cdots, J^1_{k_n}\ ]$ e $S^2 = [\ J^2_{k_1}, J^2_{k_2}, \cdots, J^2_{k_n}\ ]$

Se S^1 e S^2 forem atribuídos a duas máquinas idênticas, funcionando em paralelo, com velocidade V, o momento de conclusão de cada tarefa é idêntico nos dois sistemas.

Demonstração:

Seja C_{k_i} o tempo de conclusão da tarefa J_{k_i} num sistema da processamento com uma máquina a velocidade 2V. C_{k_i} virá então:

$C_{k_i} = \sum_{j=1}^{i} (p_{k_j} + I_{k_{j-1}, k_j})$, onde I_{k_{j-1}, k_j} é o tempo de inactividade da máquina entre o fim da tarefa realizada na posição k_{j-1} e a tarefa começada na posição k_j. Para $k_{j-1}=0$, I_{k_{j-1}, k_j} é o tempo decorrido até ao início do processamento da primeira tarefa.

Supondo que cada intervalo de inactividade é igualmente uma tarefa, então também se pode repartir, como cada tarefa $[\ J_{k_1}, J_{k_2}, \cdots, J_{k_n}\ ]$, pelas duas máquinas, a velocidade V, obtendo-se:

[24] Neste caso basta que cada lote tenha um número de cartas, D, par.

$$\left.\begin{aligned} C^1_{k_i} &= \sum_{j=1}^{i}\left[\frac{1}{2}(p_{k_j} + I_{k_{j-1},\,k_j})\cdot 2\right] = C_{ki} \\ C^2_{k_i} &= \sum_{j=1}^{i}\left[\frac{1}{2}(p_{k_j} + I_{k_{j-1},\,k_j})\cdot 2\right] = C_{k_i} \end{aligned}\right\}$$ com duas máquinas. ■

Uma questão que surge, e que se procura analisar de imediato, é de se um escalonamento construído originalmente para duas máquinas que operam em paralelo, não atingirá um valor substancialmente melhor para a função de desempenho.

4.5.1.Duas máquinas a velocidade V *versus* uma máquina a velocidade 2V

4.5.1.1.Escalonamento *S* para uma máquina a partir de *S'* para duas máquinas

No que se segue e duma forma construtiva, prova-se que nas condições deste problema (4.1. e 4.2.), sendo S' um escalonamento estabelecido para duas máquinas iguais a funcionar em paralelo, isto é onde as tarefas possam ser realizadas numa de duas máquinas com velocidade de processamento V, é possível obter de S' um escalonamento S, para uma máquina com velocidade 2V, e nas mesmas condições restantes de S', com $n_T(S) \leq n_T(S')$.

Lema L2: Seja A um conjunto finito de números reais, $a_1, a_2, \ldots, a_n$ e faça-se uma partição de A nos subconjuntos B e C.

Seja E_B a soma dos elementos de B, E_C a soma dos elementos de C e $E=\text{Max}\{E_B, E_C\}$.

Nestas condições, a semi-soma dos elementos de A não é superior a E. Formalizando:

$$\sum_{a_i \in A} \frac{a_i}{2} \leq \text{Max}\left\{ \sum_{a_i \in B} a_i \, , \sum_{a_i \in C} a_i : B \cap C = \varnothing \wedge B \cup C = A \wedge a_i \in \Re \right\}$$

Demonstração:

1.Suponha-se que o resultado não é válido. Então:

$$\sum_{a_i \in A} \frac{a_i}{2} > \text{Max}\left\{ \sum_{a_i \in B} a_i \, , \sum_{a_i \in C} a_i \right\} \Rightarrow \begin{cases} \sum_{a_i \in A} \frac{a_i}{2} > \sum_{a_i \in B} a_i \\ \sum_{a_i \in A} \frac{a_i}{2} > \sum_{a_i \in C} a_i \end{cases} \Rightarrow$$

$$\Rightarrow \sum_{a_i \in A} \frac{a_i}{2} + \sum_{a_i \in A} \frac{a_i}{2} > \sum_{a_i \in B} a_i + \sum_{a_i \in C} a_i \Rightarrow \sum_{a_i \in A} a_i > \sum_{a_i \in A} a_i$$

2.Resulta então de 1., por redução ao absurdo, que:

$$\sum_{a_i \in A} \frac{a_i}{2} \leq \text{Max}\left\{ \sum_{a_i \in B} a_i \, , \sum_{a_i \in C} a_i : B \cap C = \varnothing \wedge B \cup C = A \wedge a_i \in \Re \right\}.$$

Proposição P2: Seja S' um escalonamento elaborado para minimizar o número de tarefas em atraso, a serem processadas num sistema de duas máquinas em paralelo, isto é as tarefas passam obrigatoriamente por uma de duas máquinas e nunca pelas duas, e nas restantes seguintes condições:

- Velocidade de processamento: V;
- O tempo total de funcionamento do sistema pode ser dividido em L períodos de duração Δ_t, de tal forma que qualquer tarefa seja começada e concluída num só período i ($1 \leq i \leq L$).

- Qualquer tarefa executada no período i, tem *ready time* não superior ao limite inferior do período i: $t_i = i \times \Delta_t$.

Então existe um escalonamento S, para uma só máquina com velocidade 2V, tal que $n_T(S) \leq n_T(S')$.

Demonstração:

Considerem-se as tarefas não atrasadas, executadas com S' no período i ($1 \leq i \leq L$), e ordenem-se por tempos de conclusão, $C_1 \leq C_2 \leq ,..., \leq C_n$.

Tenham $[J_1, J_2, ..., J_n]$, em S', os tempos de execução $p_1(S'), p_2(S'), ..., p_n(S')$.

Admita-se que em S são realizadas no mesmo período i e na mesma ordem de conclusão de S'.

Tome-se agora uma tarefa não atrasada concluída em S', J_u. Para tempo de conclusão de J_u em S' vem:

$C_u(S') \geq t_i + \sum_{k \in k1} p_k(S')$ [25], sendo k1 o conjunto formado pelas tarefas não atrasadas processadas numa das máquinas (sem perda de generalidade M_1), onde J_u é realizada, que antecedem J_u adicionado de J_u.

Seja k2 o conjunto de tarefas não atrasadas processadas em M_2, concluídas não posteriormente a J_u. Então:

$C_u(S') \geq t_i + \sum_{k \in k2} p_k(S')$, e como se mantém a ordem de conclusão em S, vem para tempo de conclusão da tarefa J_u em S:

$$C_u(S) = t_i + \sum_{k \in (k1 \cup k2)} \frac{p_k(S')}{2} \leq t_i + \text{Max}\left\{ \sum_{k \in k1} p_k(S'), \sum_{k \in k2} p_k(S') \right\},$$

por L2 e daqui portanto $C_u(S) \leq C_u(S')$.

Como J_u não está atrasada em S', então também o não está em S, e igualmente para $\forall$ J_k, $1 \leq k \leq n$, concluída sem atraso no período i em S'.

Do anterior conclui-se a veracidade de P2, para um período genérico i e, consequentemente, para todos os períodos i = 1, 2, ..., L.

4.5.1.2.Escalonamento *S'* para duas máquinas a partir de *S* para uma máquina

Recupere-se agora o problema anterior, de se o método descrito em 4.3. for um "bom escalonamento", *S*, para um sistema com uma máquina cuja velocidade é 2V (e mantendo todas as condições que se vêm admitindo), é igualmente "bom" se for adaptado, na forma referida em 4.5., a um sistema de duas máquinas em paralelo à velocidade V.

Sejam *S* (obtido com 4.3.), e *S'*, escalonamentos, respectivamente, para sistemas com uma máquina à velocidade 2V, e duas máquinas em paralelo, à velocidade V.

Representem-se os números de lotes em atraso para escalonamentos óptimos para "uma máquina" por n_T^{1M} e para "duas máquinas" por n_T^{2M}.

Ora por P2, determinado *S'*, consegue-se um escalonamento para uma só máquina com um número de cartas em atraso não superior, concluindo-se, portanto, que $n_T^{2M} \geq n_T^{1M}$.

---

[25] Se não se verificarem tempos de inactividade entre tarefas e as tarefas processadas em tempo forem sucessivas, ter-se-á: $C_u(S') = t_i + \sum_{k \in k1} p_k(S')$

Admita-se que S garante um valor óptimo para a função de desempenho, $n_T^{1M}(S)$.

Então $n_T^{2M}(S') \geq n_T^{2M} \geq n_T^{1M}(S)$, isto é, aplicando S a duas máquinas obtém-se igualmente o melhor resultado possível.

Passando agora à transposição destes resultados para o problema que se tem vindo a estudar, considere-se S' um escalonamento para duas máquinas em paralelo com velocidade V, sendo D a dimensão escolhida para os lotes a processar.

O total de cartas processadas antes da hora de corte é então: D × número de lotes divididos sem atraso.

Seja S um escalonamento gerado segundo o algoritmo 4.3. para lotes de dimensão, também, D. Como, por 4.4., S é óptimo então, o número de lotes divididos em tempo segundo S; é não inferior aos divididos em tempo segundo S'.

Empregando agora o método exposto em 4.5., de fraccionar ao meio os lotes de dimensão D escalonados segundo S distribuindo-os pelas duas máquinas, facilmente se conclui que o total de cartas divididas em tempo, gerando um escalonamento para uma máquina fictícia a velocidade 2V, é superior ou igual ao total das divididas a partir de um escalonamento S' gerado para duas máquinas reais em paralelo a velocidade V.

4.6.Uma máquina e prioridades diferenciadas

Coloque-se agora, a possibilidade de prioridades diferentes para as várias saídas dos *OCR*.

Seja j uma saída dos *OCR* considerada prioritária relativamente a uma saída k, ($1 \leq j \leq m$; $1 \leq k \leq m$). Pretende-se um escalonamento que não permita processar lotes de k num período i, se existirem lotes em atraso de j que possam ser divididas em i, e se a substituição de lotes de k por lotes de j em i, diminuir o número de lotes em atraso de j.

O modo como se propõe atingir este objectivo é complementar, com uma segunda fase, o escalonamento resultante de 4.3., com a execução do próximo procedimento, sem alterar o número de lotes não divididos até à hora de corte.

4.6.1.Complemento do procedimento 4.3.

1.Estabelecer um escalonamento em obediência às etapas enumeradas em 4.3.;

Defina-se $T_j(i)$, como tráfego sobrante da saída j no período i, igual a:

lotes de j chegados até ao início do período i - lotes divididos de j até ao fim do período i.

2.Ordenar, decrescentemente, as saídas dos *OCR* por prioridades;

3.Fazer j=1;

4.Tomar a saída j com prioridade p(j);

5.Determinar o período, i, correspondente à sua hora de corte.

6.Se $T_j(i)= 0$, saltar para 15;

7.Verificar se em i existe alguma saída, k, a ser processada (em tempo inferior a hora de corte de j) e menos prioritária ;

8.Se não existir saltar para 14;

9.Observar o intervalo de tempo em que k está a ser tratada em i, Δ_i;

10.Determinar $\Delta_1 = \min \{ T_j(i) \cdot D/V, \Delta_i \}$, onde V é a velocidade da máquina de divisão,

11.Se $\Delta_1 = \Delta_i$, substituir integralmente os lotes de k por lotes de j; caso contrário trocar apenas os equivalentes ao intervalo de tempo necessário para dividir $T_j(i)$ sem atraso.

12.Adicionar, para todos os períodos, q, $i \leq q \leq L$, o tráfego de k, resultante de 11, que passou a não tratado, $T_k(q)$ (tráfego sobrante de k);

13.Actualizar para o período i o valor de $T_j(i)$ para $T_j(i) - \Delta_1 \cdot V/D$.

Voltar a 6;

14.Fazer i=i-1. Se i=0 saltar para 15;

Actualizar para o período i (i>1) o valor de $T_j(i)$ para $\min \{ T_j(i), T_j(i+1) \}$ (lotes de j que faltam tratar até ao período i).

Voltar a 6.

15.Fazer j=j+1. Se j < m voltar a 4;

Uma pequena correcção é agora necessária, para que os lotes de qualquer saída dos *OCR* sejam divididos continuamente em qualquer intervalo, e se transportar o escalonamento obtido para a situação de duas máquinas como explicada em 4.5..

Tome-se, como exemplo do que se pretende dizer, um período i onde sejam divididos lotes das saídas l, m e k, nesta ordem, e seja j, uma saída com lotes em atraso que possam ser divididos em i. Admita-se, ainda, que a prioridade de j, é superior às de l e k, mas inferior a m.

Aplicando o procedimento anterior no período i, poderiam ficar lotes de j seguidos de lotes de m e novamente lotes de j. O que se teria de fazer para o evitar era deslocar m para o lugar dos primeiros lotes de j (na anterior posição de l) e, consequentemente, estes lotes de j para o primitivo lugar de m, de forma a que todos os lotes de j fiquem contíguos e processados os lotes de ambas as saídas, m e j, antes do tempo limite.

CAPÍTULO 5

GERAÇÃO DE SOLUÇÕES PARA O PROBLEMA DA ALIMENTAÇÃO DAS DIVISORAS COM BASE NUM MODELO DE OPTIMIZAÇÃO DE ROTAS

Mas o que a eles não toca
É a magia que evoca
O Longe e faz dele história.

(...)

Fernando Pessoa, Mensagem

5.1. Descrição do problema

Admita-se que quando a alimentação das divisoras é mudada de lotes da saída j para lotes da saída q ($q \neq j$), é necessário um período de tempo não nulo para proceder à substituição, sendo este intervalo de tempo igual a zero se $q = j$.

Seguindo [French, 1992] tome-se como tempo de execução de uma tarefa o tempo de divisão de um lote adicionado ao tempo de *set-up* da divisora. Então o tempo de processamento é função da forma como os lotes são sequenciados nas divisoras.

Em [Lawler et al, 1993] é feito notar que escalonamentos com tempos de processamento dependentes da sequência com que as tarefas transitam nas máquinas do sistema de processamento (*sequence-dependent processing times*), são afins aos problemas do caixeiro viajante e suas extensões.

Dando continuidade a esta ideia, vai ser tema deste capítulo a resolução do problema completo enunciado em 2.4.2.2., com um modelo de optimização de rotas de veículos (*VRP - Vehicle Routing Problem*).

O problema da determinação de rotas de veículos pode ser introduzido com o seguinte enunciado [Eilon et al, 1971]:

Estabelecer os percursos de uma frota de veículos que parte de um depósito para satisfazer encomendas de um conjunto de clientes. As localizações do depósito e clientes, a capacidade dos veículos e as quantidades e produtos a fornecer são dados do problema; as restrições são as que se enumeram:

- As encomendas de todos os clientes são satisfeitas;
- A capacidade dos veículos, nomeadamente em peso e volume, não é ultrapassada;
- O tempo, ou a distância, total gasto por cada veículo não pode exceder um valor fixado;
- Existe uma janela temporal na qual o cliente tem de ser servido.

O objectivo é minimizar o custo de entrega de todas as encomendas aos clientes, o qual se pode decompor em dois termos: custo associado com a dimensão da frota e custo para realização dos percursos.

5.2.Modelização

Considerando uma encomenda como um lote de cartas de uma saída dos *OCR*, as semelhanças entre o problema do escalonamento, na forma completa exposta em 2.4.2.2., e o *VRP* começam por manifestar-se na janela temporal que ambos observam para uma acção: no caso do *VRP* para fazer a entrega de uma encomenda, no escalonamento para efectuar a divisão de um lote. Um segundo ponto comum pode estabelecer-se com a obrigatoriedade de tempos de preparação nas divisoras. Admitindo que

cada cliente é uma saída dos *OCR*, quando a alimentação das máquinas divisoras deixa de ser com lotes da saída j e passa a ser com lotes da saída q, e se pretende que isto origine um espaço de tempo em que não se pode processar correio para a máquina ser esvaziada, então é como se tivesse que ir do cliente j para o cliente q, motivando com isso uma ocupação do veículo ao efectuar a viagem entre os locais j e q.

Um veículo, na forma genérica em que foi apresentado o problema de geração de rotas, parece ocupar apenas o tempo com deslocações. Ora fazendo a associação máquina de divisão ↔ veículo é necessário incluir, para além do tempo de preparação, o tempo gasto na divisão das cartas. Embora não explicitado em 5.1., é possível considerar um tempo para carga ou descarga das encomendas. Levanta-se aqui a possibilidade de uma nova ponte entre os dois problemas se se fizer do tempo de processamento de um lote numa divisora o equivalente ao tempo de descarga de uma encomenda num cliente.

Quando se fixa o número de veículos pode ser impossível satisfazer todas as encomendas dentro das janelas temporais exigidas pelos clientes. É então objectivo gerar rotas de modo a satisfazer o máximo de pedidos em tempo.

Imaginando a marcação de cada lote de cartas proveniente das saídas dos *OCR* para identificação das "encomendas a levar ao respectivo cliente", obter-se-ão as rotas com a informação: o veículo (n) entrega a encomenda (e) no cliente (j) no momento (h), que interpretado na perspectiva do problema do escalonamento se traduz em colocar na divisora (n) um lote da saída (j) dos *OCR* no instante (h).

Terminada esta nota introdutória, passa-se de imediato ao detalhe das correspondências entre os dois problemas, restando apenas sublinhar que sendo, como já dito no capítulo 1, a tarefa computacional realizada recorrendo a uma aplicação comercial

(Optrak), as analogias entre o *VRP* e o problema do escalonamento que se pretende resolver, serão guiadas pelo modelo de *VRP* em que o Optrak se fundamenta.

5.2.1.Clientes

Os clientes são considerados como as saídas 1,2,..., m dos *OCR*.

5.2.2.Rede de estradas e localização dos clientes

5.2.2.1.Rede/localização

A rede de caminhos transitáveis tem como nós, ou vértices, m+1 "locais"; os m clientes e um ponto Ω que representa o depósito.

A rede é formada com a particularidade de ser um grafo completo, isto é, todos os nós são adjacentes. A existência de uma ligação directa entre os clientes j e q e do cliente j para o depósito Ω (j, q=1,2,...,m), garante que se pode mudar de uma saída dos *OCR* para outra saída sem passar por mais nenhum ponto, e que pode iniciar-se e finalizar-se o processo de divisão com um lote de qualquer saída.

- movimento entre dois nós é permitido em ambos os sentidos e a distância que separa qualquer par de nós tem o mesmo valor: Dist.

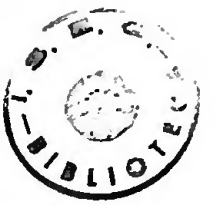
5.2.2.2. Velocidade

- Se o problema não considera tempos de esvaziamento, a velocidade num arco em que ambos os vértices são clientes j e q (v_{jq}), é escolhida de modo a que $\frac{Dist}{v_{jq}} \approx 0$, isto é, não se desperdiça tempo quando se muda a alimentação da divisora de lotes da saída j para lotes de outra saída q.
- Se o problema considera tempos de esvaziamento, a velocidade num arco em que os extremos são os clientes j e q é atribuída de forma a que $\frac{Dist}{v_{jq}}$ ("desempenhando o papel" de tempo de percurso entre os dois clientes), seja o tempo consumido para esvaziar a divisora de lotes da saída j e prepará-la para tratar lotes da saída q.

  Pretendendo que a mudança de q para j ocupasse tempo diferente da de j para q bastaria ter um arco (j,q) e outro (q,j) com velocidades (ou distâncias) diferentes. No caso presente tal não se afigura necessário e além disso $v_{jq} = v$, $\forall$ j, q.
- A velocidade num arco em que um dos vértices é o depósito é $\frac{v}{4}$, para evitar que as ligações entre clientes passem pelo depósito.

  A redução da velocidade entre depósito e clientes, ou o inverso, poderão levar a pensar que se está a inutilizar tempo nos "trajectos" inicial (do depósito para o primeiro cliente), e final (de regresso ao depósito), que se poderia usar para dividir mais lotes.

  Contudo, a "distribuição" pode começar de maneira a que a chegada ao primeiro cliente coincida com o começo do período de funcionamento do sistema e terminar suficientemente depois da última hora de corte.



5.2.3.Encomendas

Os produtos a fornecer aos clientes são os lotes de cartas a processar nas divisoras, originados nas saídas respectivas dos *OCR*, sendo cada lote uma encomenda.

Fixe-se uma dimensão, D, e calcule-se como em 4.2. o número de lotes que existe para dividir em cada intervalo i e por saída j dos *OCR*: J_i^j. Então, supondo que todos os lotes estão no depósito, é preciso entregar ao cliente j $\sum_{i=0}^{L-1} J_i^j$ encomendas.

Ficam, assim, conhecidos o tipo de produto e a quantidade que cada cliente tem que receber.

5.2.4.Veículos

O número de veículos que compõe a frota é igual ao número de máquinas divisoras existentes em cada CTC.

Admite-se que os veículos têm capacidade de carga ilimitada e não se impõe um máximo para a distância ou tempo de percurso.

5.2.5.Janela temporal

Um lote da saída j dos *OCR* que chega no período i tem que ser entregue no cliente j no intervalo $\left[(i+1)\cdot\Delta_t\ ,\ h_c - \frac{D}{V}\right]$[26], onde h_c é a hora de corte para os lotes da saída j, D a dimensão dos lotes e V a velocidade das máquinas de divisão em cartas/hora.

5.2.6.Outros factores

Embora não explicitados em 5.1. são para este caso, como se viu, relevantes:

5.2.6.1.Tempo de descarga de cada encomenda

Dispende-se um tempo $\frac{D}{V}$ horas para entregar um lote em qualquer cliente. Funciona, como se disse, como um período de descarga no qual o veículo está imobilizado, significando para este caso ocupar a divisora no tempo necessário para processar um lote. Condiciona-se, assim, a quantidade de lotes que é possível dividir em cada hora à velocidade de processamento dos equipamentos.

---

[26]Quando $(i+1)\cdot\Delta_t < h_c - \frac{D}{V}$

5.2.6.2.Prioridades

A entrega a um cliente j, pode ser preferencial relativamente a um cliente q. Tal equivale a dizer que existem prioridades diferenciadas para processar os lotes.

5.2.7.Função objectivo

Em vez de minimizar o custo para satisfazer todos os pedidos de encomendas, procuram-se "rotas", ou seja um escalonamento que minimize o número de encomendas não entregues, correspondendo a minimizar o número de lotes em atraso.

CAPÍTULO 6

GERAÇÃO DE SOLUÇÕES PARA O PROBLEMA DA ALIMENTAÇÃO DAS DIVISORAS COM BASE NUM MODELO DE AFECTAÇÃO

(...)

É por isso que a sua glória
É justa auréola dada
Por uma luz emprestada.

Fernando Pessoa, Mensagem

6.1.Descrição do problema

Em [Brucker, 1995], é referido que problemas de escalonamento da classe $1 \mid r_k, p_k = 1 \mid \sum_k f_k(C_k)$, sendo $f_k(C_k)$ regular, podem ser formulados como problemas de afectação. Concretizando esta abordagem, apesar dos inconvenientes apontados, vai ser intuito deste capítulo dar-lhe corpo com o caso presente e alargando-a para um sistema de processamento com M máquinas em paralelo, correspondente à totalidade das máquinas de divisão existentes nos três CTC. Procura-se, assim, conceptualizar um escalonamento integrado para o conjunto dos Centros de Tratamento de Correio, obtendo-se um modo de avaliar se a permuta de correio inter-CTC resulta num incremento do número de cartas divididas antes da hora limite respectiva, não considerando tempos de esvaziamento.

O problema da afectação é um caso particular da programação linear. Uma forma de apresentação do seu âmbito e objectivos, utilizada por vários autores, [Ackoff et al, 1968], [Guerreiro et al, 1984], [Tavares et al, 1996], é com o exemplo da atribuição de n tarefas a n trabalhadores, de tal forma que cada trabalhador só pode

executar uma tarefa e cada tarefa só pode ser realizada por um trabalhador. São conhecidos os custos de desempenho de cada trabalhador em cada tarefa e pretende-se afectar cada um dos n trabalhadores a uma e só uma das n tarefas, de modo a que todas as tarefas sejam cumpridas com custo total mínimo.

Considerando que $x_{ij} = 1$ significa que o trabalhador i é encarregue da tarefa j e $x_{ij} = 0$ no caso contrário e que c_{ij} é o custo do trabalhador i quando colocado na tarefa j, o problema pode formalizar-se no modo seguinte:

$$\text{Min } F=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, x_{ij} \ ,$$

sujeito a:

$$\sum_{j=1}^{n} x_{ij} = 1, (i = 1,2,\ldots,n)$$

$$\sum_{i=1}^{n} x_{ij} = 1, (j = 1,2,\ldots,n)$$

$$x_{ij} = 1, \text{ ou } x_{ij} = 0, \ (i, j = 1,2,\ldots,n)$$

Embora este tipo de problema possa ser resolvido como um caso de programação linear (apesar da restrição que obriga as variáveis a só poderem assumir os valores 0 ou 1, esta pode substituir-se pela condição de não negatividade, vulgar em problemas de programação linear, sem que a solução óptima deixe de corresponder a inteiros nulos ou unitários), a sua estrutura torna possível o desenvolvimento de algoritmos específicos, nomeadamente o chamado método húngaro, o que lhe confere individualidade.

6.2.Modelização

Considerem-se os L períodos de funcionamento de um CTC, tal como no capítulo 4. Formem-se lotes de dimensão D com as cartas chegadas em cada intervalo $[i\Delta_t, (i+1)\,\Delta_t\,[$, para cada saída j dos *OCR* , Q_i^j ($1 \le i < L$; $1 \le j \le m$), e com as cartas recebidas até ao início do primeiro período, igualmente, para cada saída j, Q_0^j. Até ao início do primeiro período existem, então, para cada saída j: $J_0^j = \left\lfloor \frac{Q_0^j}{D} \right\rfloor$ lotes. O resto da divisão inteira é considerado, como anteriormente, integrado no intervalo seguinte. Procedendo de forma semelhante para todos os restantes intervalos obtêm-se, deste modo, mais J_i^j ($1 \le i < L$; $1 \le j \le m$) lotes. As cartas recebidas até ao começo do primeiro período podem ser divididas em qualquer período i, as que são recepcionadas em $[i\Delta_t, (i+1)\,\Delta_t\,[$ ($1 \le i < L$), podem entrar nas máquinas divisoras a partir do início de i+1.

Subdivida-se agora cada período i em partes iguais criando, i_1, i_2,..., i_K, subperíodos, onde a duração de i_l ($1 \le l \le K$), seja equivalente ao tempo consumido na divisão de um lote de D cartas. A contagem do tempo é, como atrás, feita a partir do começo do primeiro período.

Fazendo agora a analogia com os trabalhadores e as tarefas, os lotes são os trabalhadores e os subintervalos de tempo $\left[\Delta_t i + \frac{\Delta_t}{K}(l-1), \Delta_t i + \frac{\Delta_t}{K} l\right[$, ($1 \le i \le L; 1 \le l \le K$), as tarefas,[27] isto é, pretende-se afectar lotes a subperíodos de funcionamento.

[27] O número de subintervalos em cada período i é igual ao número de máquinas divisoras no CTC×K.

Analise-se agora a correspondente matriz de custos associada a cada lote d, proveniente da saída j, para o afectar a um subperíodo s:

- Se s está entre a chegada e a hora de corte de d, isto é, $s \subseteq [r_d, d_d]$, então o custo é mínimo: $c_{ds} = c_{min} = 0$;
- Se $s \not\subset [r_d, d_d]$, então o custo é o peso atribuído à saída j (j =1,...,m), dos *OCR* a que o lote pertence: $c_{ds} = w_j > 0$;

Como o número de lotes tem que ser igual ao número de subperíodos, poderá tornar-se necessário incluir subperíodos, ou lotes, fictícios.

- O custo de afectar um lote d num subperíodo fictício é w_j;
- O custo de afectar um lote fictício num subperíodo s ou num subperíodo fictício é cf.

Minimizar a soma dos custos totais, significa então minimizar a soma ponderada de todos os lotes divididos com atraso, já que o custo associado ao lotes fictícios é fixo.

Note-se ainda que:

- Um lote d1 nunca pode ser deixado em atraso se houver um lote d2 (ou um lote fictício), afecto a um subperíodo s: $s \subseteq [r_{d1}, d_{d1}] \wedge s \not\subset [r_{d2}, d_{d2}]$.

  Então trocando d1 e d2 (ou um lote fictício) de subperíodos obtinha-se uma variação negativa do custo total: $(c_{min} + w_{d2}) - (w_{d1} + w_{d2})$, ou $(c_{min} + cf) - (w_{d1} + cf)$;
- Um lote d1 não pode ficar em atraso se existir d2 (ou um lote fictício), afecto a s2: $s2 \subseteq [r_{d1}, d_{d1}]$ e $w_{d1} > w_{d2}$.

  Novamente, trocando d1 e d2 de subperíodos obtinha-se também, uma variação negativa do custo total: $(c_{min} + w_{d2}) - (w_{d1} + c_{min})$, ou $(c_{min} + w_{d2}) - (w_{d1} + w_{d2})$, ou $(c_{min} + c_f) - (w_{d1} + c_f)$.

Concluir-se-ia então, em qualquer das situações, que era possível melhorar a função objectivo, não se estando em presença do óptimo.

6.3.Permuta inter-CTC

Como dito atrás, equacionando este caso de escalonamento como um problema de afectação torna-se possível imaginar um escalonamento integrando o conjunto de todos os CTC.

Para cartas recebidas, por exemplo, em Lisboa que tenham como destino CDP da área do CTC do Porto (ou da área do CTC de Coimbra), pode questionar-se se deverão ser indexados e divididos em Lisboa, seguindo depois para o Porto (ou Coimbra), ou se apenas deverão ser indexados no CTC de Lisboa, deixando a divisão ser realizada nos CTC mais próximos dos destinos. Isto corresponde a não olhar cada centro de tratamento isoladamente, mas como um sistema integrado onde lotes de alguns grupos de códigos poderiam, à saída dos *OCR* onde fossem indexados, ser encaminhados para finalizar o processo de tratamento em qualquer dos três centros, tentando melhorar a capacidade conjunta das quatro divisoras.

Admita-se agora que se atribuem saídas dos *OCR*, em cada CTC, para estes grupos de destinos de que importa avaliar os potenciais benefícios da permuta inter-CTC.

Relembrando 2., repete-se que existem carreiras entre os CTC para transporte de correio, o que significa que um destes lotes chegado, por exemplo, ao CTC de Lisboa no período i com códigos de destino da região do CTC do Porto, está disponível para ser dividido no CTC de Lisboa em $(i+1)\cdot\Delta_t$ e no CTC do Porto na hora a

que a carreira Lisboa → Porto imediatamente mais próxima de $(i+1)\cdot\Delta_t$ chega ao Porto (excluindo-se a possibilidade de ser processado em Coimbra).

O *ready time* para estes lotes, ao contrário dos *due time* é, evidentemente, superior se forem triados nos CTC mais próximos dos CDP de destino.

Das considerações anteriores, é importante observar que não se trata exactamente de um problema da classe $PM \mid r_k, p_k = 1 \mid \sum_k w_k U_k$, pois os *ready times* e os *due times* dependem da máquina (ou CTC) onde os lotes são processados.

Voltando agora ao modelo de afectação, podem considerar-se 4 divisoras utilizáveis. Como a máquina de divisão de Coimbra tem 2/3 da velocidade das de Lisboa e Porto, fixando K subintervalos em Lisboa e Porto, resulta um total de $3K + \frac{2}{3}K$ subintervalos em cada período i[28] onde se podem afectar lotes de dimensão D. A figura seguinte elucida o que se pretende afirmar:

[28] Escolhendo um valor para D de modo a que (2/3)K seja inteiro.

Figura 5

Subintervalos possíveis para afectação dos lotes

Intervalo i

	Subintervalo		
	1	.....	**K**
Divisora 1 de Lisboa	$\frac{\Delta_t}{K}$	.....	$\frac{\Delta_t}{K}$
Divisora 2 de Lisboa	$\frac{\Delta_t}{K}$	.....	$\frac{\Delta_t}{K}$
Divisora do Porto	$\frac{\Delta_t}{K}$	.....	$\frac{\Delta_t}{K}$

Intervalo i

	Subintervalo		
	1	.....	$\frac{2}{3}$ **K**
Divisora de Coimbra	$\frac{3}{2}\frac{\Delta_t}{K}$	.....	$\frac{3}{2}\frac{\Delta_t}{K}$

Um lote recebido no CTC_k com CDP do CTC_k é possível ser assignado a subintervalos entre a sua hora de chegada e a hora de corte (no CTC_k), numa divisora do CTC_k com custo nulo e fora dele com custo superior, mas nunca deverá ser processado no CTC_n ($n \neq k$).

De forma análoga a 6.2. a matriz de custos virá então:

- Lotes recebidos no CTC_k da saída j no período i com destinos do CTC_k:

- assignados a subintervalos entre $(i+1)\cdot\Delta_t$ e a hora de corte no CTC_k, em divisoras do CTC_k: c_{min};

- assignado a outros intervalos do CTC_k: w_j;

- outras situações: ∞

- Lotes recebidos no CTC_k da saída j no período i com destinos do CTC_n:

  - assignados a subintervalos entre $(i+1)\cdot\Delta_t$ e a hora de corte no CTC_k, em divisoras do CTC_k: c_{min}.

  - assignados a subintervalos posteriores à sua chegada ao CTC_n e a hora de corte no CTC_n, em divisoras do CTC_n: c_{min};

  - assignados a outros intervalos no CTC_k ou no CTC_n,: w_j;

  - outras situações: ∞.

Analisando os resultados, torna-se possível concluir se a abertura deste intercâmbio entre os CTC permite aumentar o número de lotes divididos sem atraso.

CAPÍTULO 7

RESULTADOS COMPUTACIONAIS

Diz-me quem és e donde vens
Conta-me lá os teus feitos
Que eu nunca vi pátria assim
Pequena e com tantos peitos

Carlos Tê, A valsinha das medalhas

A aplicação dos métodos descritos anteriormente foi realizada sobre o CTC de Lisboa, com maior volume de tráfego e informação mais actualizada (1995).

O quadro 1 mostra o diagrama de cargas relativo às cartas recebidas, horas de corte e prioridades (uma saída com o valor u é mais prioritária que outra com o valor v < u), por saída dos *OCR*. Os agrupamentos de destinos em cada saída, as horas de corte e prioridades são arbitradas, dado que à data de realização da dissertação ainda se estava na fase de instalar os *OCR*. Os valores de tráfego, atendendo aos agrupamentos, são factuais.

Quadro 1

CTC de Lisboa: digrama de cargas, horas de corte e prioridades por saída dos *OCR* (valores estimados em 1995)

Período ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Período	Hora	Prioridade
Hora ->	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	de corte	de corte	
Saídas *OCR*																											
1	9357	6908	6294	7041	21748	5440	16733	10557	12665	11485	6028	1044	1512	3163	2291	993	2456	2659	2298	1198	771	1609	2463	15904	16	3H:00M	10
2	3084	2291	2143	2374	7040	1779	5545	3529	4147	3808	2012	340	538	1022	726	331	720	864	783	422	256	559	759	5141	17.4	4H:24M	8
3	8663	6453	5836	6298	19700	5133	15011	9552	11281	10530	5442	973	1340	2955	2037	920	2099	2385	2116	1135	749	1543	2188	14624	19	6H:00M	8
4	4083	3022	2841	3044	9451	2431	7173	4599	5383	5078	2728	446	684	1394	930	437	966	1147	979	555	351	665	1051	7102	9.4	20H:24M	8
5	3843	2881	2618	2947	8994	2340	6963	4370	5011	4759	2429	460	633	1259	963	396	990	1150	917	532	317	673	1015	6581	14.6	1H:36M	8
6	5080	3789	3472	3848	11872	2947	9180	5769	6671	6298	3202	583	756	1727	1158	553	1300	1465	1253	667	414	889	1311	8676	13	24H:00M	1
7	6694	5028	4554	4916	15063	3858	11684	7383	8714	8258	4220	780	1104	2228	1592	665	1610	1887	1611	899	573	1231	1691	11328	14	1H:00M	10
8	2174	1629	1497	1638	5000	1291	3831	2523	2962	2669	1416	262	363	754	527	237	523	593	555	315	187	381	557	3725	16.8	3H:48M	9
9	5328	4076	3574	4002	12358	3100	9538	6119	7076	6845	3368	596	817	1800	1303	581	1368	1576	1334	662	431	1004	1380	9141	17.8	4H:48M	8
10	1866	1326	1338	1420	4355	1104	3317	2018	2440	2275	1141	186	265	634	474	218	445	540	486	241	154	329	529	3173	13.4	0H:24M	7
11	12751	9476	8895	9546	29396	7401	22968	14134	16887	15697	8077	1469	2035	4363	3138	1338	3079	3735	3172	1599	1038	2264	3392	21899	12.4	23H:24M	6
12	2344	1725	1630	1765	5458	1274	4161	2525	3038	2843	1530	283	387	796	561	239	543	646	578	293	178	404	611	4081	15.4	2H:24M	5
13	10596	8162	7408	7872	24665	6462	18846	12064	14128	13409	6773	1255	1715	3626	2558	1103	2664	3027	2540	1440	882	1883	2731	18269	15	2H:00M	4
14	30366	22631	20661	22375	69814	17671	53216	33885	39897	37605	18975	3338	4846	10178	7301	3127	7458	8750	7421	3998	2522	5461	7940	51751	17	4H:00M	3
15	3370	2559	2271	2582	7761	1967	5791	3862	4427	4242	2095	354	533	1187	812	357	844	954	853	442	288	574	835	5903	17.8	4H:48M	2

7.1.Soluções geradas com o procedimento 4.3.

Para ilustração do método, programou-se o procedimento 4.3. em linguagem C para DOS.

Permite que sejam variáveis o número de períodos, o número de saídas dos *OCR* e a dimensão/número dos lotes, sendo os máximos possíveis em grandeza bastante superior às necessidades práticas. A entrada e saída de dados é feita através de ficheiros em formato ASCII .

A escolha do CTC de Lisboa, com duas máquinas a funcionar em paralelo[29], para incidir nele a aplicação do procedimento 4.3., não permite a aplicação directa do escalonamento gerado; mas, como se viu, pode fazer-se uma abordagem em que primeiro se utiliza o algoritmo desenvolvido considerando que, virtualmente, se opera com uma só divisora com a particularidade de a velocidade ser duplicada para 60000 cartas/hora. Posteriormente, com base nos resultados obtidos, e como se justificou em 4.5. poderá idealizar-se um escalonamento para o caso das duas divisoras em paralelo.

O quadro 2 sumaria os totais obtidos para lotes que são submúltiplos, em tempo equivalente de processamento, de um período de uma hora e na condição de prioridades iguais. As percentagens referem-se à relação entre cartas chegadas e cartas divididas sem atraso, para cada saída dos *OCR*.

[29] Coimbra e Porto têm apenas uma máquina de divisão.

Quadro 2

Cartas divididas sem atraso por dimensão dos lotes
Uma máquina, sem tempos de esvaziamento e prioridades iguais

Lotes	1000		5000		10000		15000		20000		30000	
Saídas OCR	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%
1	122000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8
2	24000	47.8	15000	29.9	0	0	15000	29.9	20000	39.9	0	0
3	72000	51.8	75000	54.0	100000	72.0	90000	64.8	80000	57.6	90000	64.8
4	42000	63.2	40000	60.1	40000	60.1	30000	45.1	40000	60.1	30000	45.1
5	49000	78.5	50000	80.1	50000	80.1	45000	72.1	40000	64.1	30000	48.1
6	62000	75.6	60000	73.2	60000	73.2	60000	73.2	60000	73.2	60000	73.2
7	82000	76.5	80000	74.6	80000	74.6	75000	70.0	80000	74.7	60000	56.0
8	28000	80.8	25000	72.1	20000	57.7	15000	43.3	20000	57.7	0	0
9	24000	27.5	30000	34.4	30000	34.4	30000	34.4	20000	22.9	30000	34.4
10	23000	76.1	20000	66.2	20000	66.2	15000	49.7	20000	66.3	0	0
11	156000	75.4	1550000	75.0	150000	72.6	150000	72.6	140000	67.8	150000	72.6
12	30000	80.0	30000	80.0	30000	80.0	30000	80.0	20000	53.3	0	0
13	136000	78.7	135000	78.1	130000	75.2	135000	78.1	120000	69.4	120000	69.4
14	230000	47.1	245000	50.2	240000	49.2	240000	49.2	240000	49.2	300000	61.5
15	0	0	0	0	0	0	0	0	0	0	0	0
Total	1080000	60.6	1080000	60.6	1070000	60.0	1050000	58.9	1020000	57.2	990000	55.5

A diminuição do número total de lotes processados observada no quadro anterior à medida que aumenta a dimensão dos lotes deve-se, como se poderá ver nos quadros que exibem os respectivos escalonamentos (quadros 3, 4, 5, A1, A2 e A3), a períodos em que a máquina fica inactiva por insuficiência de cartas para constituir lotes para os tamanhos correspondentes.

De seguida mostram-se os escalonamentos para lotes de 1000, 10000 e 20000 cartas, estando os restantes em anexo.

Quadro 3

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 1000 cartas

	Intervalos em minutos / Saídas *OCR*										
HORA											
12	0 - 4 S4	4 - 16 S11	16 -21 S6	21 - 22 S10	22 - 28 S7	28 - 31 S5	31 - 41 S13	41 - 43 S12	43 - 52 S1	52 - 54 S8	54 - 60 S14
13	0 - 3 S4	3 - 13 S11	13 -16 S6	16 - 18 S10	18 - 23 S7	23 - 26 S5	26 - 34 S13	34 - 36 S12	36 - 43 S1	43 - 44 S8	44 - 60 S14
14	0 - 2 S4	2 - 11 S11	11 -15 S6	15 - 16 S10	16 - 21 S7	21 - 24 S5	24 - 32 S13	32 - 33 S12	33 - 39 S1	39 - 41 S8	41 - 60 S14
15	0 - 3 S4	3 - 12 S11	12 -16 S6	16 - 17 S10	17 - 22 S7	22 - 25 S5	25 - 33 S13	33 - 35 S12	35 - 42 S1	42 - 43 S8	43 - 60 S14
16	0 - 10 S4	10 - 40 S11	40 - 52 S6	52 - 57 S10	57 - 60 S7						
17	0 - 2 S4	2 - 9 S11	9 -12 S6	12 - 13 S10	13 - 29 S7	29 - 40 S5	40 - 60 S13				
18	0 - 8 S4	8 - 31 S11	31 - 40 S6	40 - 43 S10	43 - 54 S7	54 - 60 S5					
19	0 - 4 S4	4 - 18 S11	18 - 23 S6	23 - 25 S10	25 - 33 S7	33 - 38 S5	38 - 60 S13				
20	0 - 6 S4	6 - 23 S11	23 - 30 S6	30 - 33 S10	33 - 41 S7	41 - 46 S5	46 - 60 S13				
21	0 - 16 S11	16 - 22 S6	22 - 24 S10	24 - 33 S7	33 - 38 S5	38 - 60 S13					
22	0 - 8 S11	8 -12 S6	12 - 13 S10	13 - 17 S7	17 - 20 S5	20 - 38 S13	38 - 59 S12	59 - 60 S1			
23	0 - 1 S11	1 - 2 S7	2 - 3 S13	3 - 60 S1							
24	0 - 1 S10	1 - 2 S7	2 - 3 S5	3 - 5 S13	5 - 34 S1	34 - 55 S8	55 - 60 S14				
1	0 - 1 S5	1 - 4 S13	4 - 5 S12	5 - 8 S1	8 - 9 S8	9 - 60 S14					
2	0 - 1 S12	1 - 4 S1	4 - 60 S14								
3	0 - 60 S14										
4	0 - 24 S2	24 - 48 S9	48 - 60 S3								
5	0 - 60 S3										

Quadro 4

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 10000 cartas

HORA	Intervalos em minutos / Saídas *OCR*					
12	0 - 10 S11	10 - 20 S13	20 - 50 S14			
13	0 - 10 S11	10 - 20 S7	20 - 30 S1	30 - 50 S14	50 - 60 S3	
14	0 - 10 S11	10 - 20 S6	20 - 30 S13	30 - 40 S1	40 - 60 S14	
15	0 - 10 S4	10 - 20 S11	20 - 30 S7	30 - 40 S13	40 - 50 S5	50 - 60 S14
16	0 - 10 S4	10 - 40 S11	40 - 50 S6	50 - 60 S10		
17	0 - 10 S6	10 - 30 S7	30 - 60 S13			
18	0 - 10 S4	10 - 40 S11	40 - 50 S6	50 - 60 S7		
19	0 - 10 S11	10 - 40 S13	40 - 60 S12			
20	0 - 10 S4	10 - 30 S11	30 - 40 S6	40 - 50 S7	50 - 60 S13	
21	0 - 10 S11	10 - 20 S10	20 - 30 S7	30 - 50 S13	50 - 60 S1	
22	0 - 10 S11	10 - 20 S6	20 - 30 S7	30 - 40 S13	40 - 60 S1	
23	0 - 60 S1					
24	0 - 30 S5	30 - 50 S8	50 - 60 S8			
1	0 - 60 S14					
2	0 - 10 S12	10 - 20 S1	20 - 30 S5	30 - 60 S14		
3	0 - 60 S14					
4	0 - 30 S9	30 - 60 S3				
5	0 - 60 S3					

Quadro 5

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 20000 cartas

	Intervalos em minutos / Saídas *OCR*		
HORA			
12	0 - 40 —	40 - 60 S14	
13	0 -20 S11	20 - 40 S14	
14	0 - 20 S13	20 - 40 S1	40 - 60 S14
15	0 - 20 S11	20 - 40 S7	40 - 60 S14
16	0 - 20 S4	20 - 40 S11	40 - 60 S6
17	0 - 20 S7	20 - 40 S5	40 - 60 S13
18	0 - 40 S11	40 - 60 S6	
19	0 - 20 S12	20 - 60 S13	
20	0 - 20 S4	20 - 40 S11	40 - 60 S7
21	0 - 20 S11	20 - 40 S10	40 - 60 S5
22	0 - 20 S6	20 - 40 S7	40 - 60 S13
23	0 - 20 S13	20 - 60 S1	
24	0 - 40 S1	40 - 60 S8	
1	0 - 60 S14		
2	0 - 20 S1	20 - 60 S14	
3	0 - 60 S14		
4	0 - 20 S2	20 - 40 S9	40 - 60 S3
5	0 - 60 S3		

Olhando para os quadros 3, 4 e 5, é notório que a opção por lotes de menor dimensão, para conseguir um maior número de cartas divididas, obriga a frequentes alterações na saída dos *OCR* que alimenta a divisora, o que é um inconveniente do ponto de vista operacional (mesmo sem considerar tempos de preparação). Há, portanto e como se disse, que encontrar um equilíbrio entre o total de cartas processadas e número de substituições de programas nas máquinas de divisão, para assegurar estabilidade, isto é, um tempo mínimo sem trocas de saída dos *OCR*.

Para apreciar o efeito das prioridades na composição dos lotes processados, o próximo quadro apresenta o resultado do complemento exposto em 4.6.1., aplicado ao procedimento 4.3.

Pode ver-se, comparando os quadros 2 e 6, que mantendo o número total de lotes divididos, lotes menos prioritários (exemplo: 6 e 14), são preteridos, sempre que possível, em relação aos de maior prioridade (exemplo: 2, 3 e 9).

Tomando lotes de menor dimensão, D=1000 ou D=5000, as saídas 2, 3 e 9 têm um grande aumento de lotes processados quando são consideradas prioridades diferenciadas, relativamente à situação de prioridades iguais. No entanto, com o crescimento do tamanho dos lotes, saídas mais prioritárias mas com menos cartas podem ser "prejudicadas" por outras de menos prioridade mas com número superior de cartas: para a saída 14, por exemplo, apesar de pouco prioritária, observa-se no quadro 6 que para D>10000 a percentagem de lotes processados é mais elevada que para $D \leq 10000$. Comparando os quadros 2 e 6 (continuando a olhar para a saída 14), e para $D \geq 10000$ quase não há alteração considerando-se ou não prioridades distintas.

Situação diversa se passa com as saídas 3 e 9 para as quais, embora se tenham percentagens de lotes tratados maiores quando se consideram prioridades diferenciadas, os valores do quadro 6 quando $D \geq 10000$ decaem relativamente a $D < 10000$.

Quadro 6

Cartas divididas sem atraso por dimensão dos lotes
Uma máquina, sem tempos de esvaziamento e prioridades diferenciadas

Lotes	1000		5000		10000		15000		20000		30000	
Saídas OCR	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%
1	122000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8
2	42000	83.7	40000	79.7	40000	79.7	30000	59.8	20000	39.9	30000	59.9
3	115000	82.7	115000	82.7	110000	79.1	105000	75.5	120000	86.3	90000	64.8
4	44000	66.2	40000	60.1	40000	60.1	30000	45.1	40000	60.1	30000	45.1
5	49000	78.5	50000	80.1	50000	80.1	45000	72.1	40000	64.1	30000	48.1
6	0	0	0	0	10000	12.6	0	0	0	0	0	0
7	82000	76.5	75000	70.0	80000	74.6	75000	70.0	80000	74.6	60000	56.0
8	28000	80.8	25000	72.1	20000	57.7	15000	43.3	20000	57.7	0	0
9	73000	83.6	70000	80.2	80000	91.7	60000	68.8	60000	68.8	60000	68.8
10	23000	76.1	20000	66.2	20000	66.2	15000	49.7	20000	66.3	0	0
11	156000	75.4	155000	75.0	150000	72.6	150000	72.6	140000	67.8	150000	72.6
12	32000	85.3	30000	80.0	20000	53.3	30000	80.0	20000	53.3	0	0
13	135000	78.1	135000	78.1	130000	75.2	135000	78.1	120000	69.4	120000	69.4
14	179000	36.6	205000	41.9	200000	40.9	240000	49.2	220000	45.0	300000	61.5
15	0	0	0	0	0	0	0	0	0	0	0	0
Total	1080000	60.6	1080000	60.6	1070000	60.0	1050000	58.9	1020000	57.2	990000	55.5

7.2.Soluções geradas com base num modelo de optimização de rotas

Para tratar problemas de optimização de rotas, os CTT adquiriram um produto, comercializado em Portugal, designado Optrak. É com este *software* que se vai resolver o problema 2.4.2.2., na sua versão completa, modelizado como um *Vehicle Routing Problem*.

A aplicação baseia-se no algoritmo de duas fases de Christofides e Mingozzi [Optmiza, 1995], do tipo *cluster first-route second*, em que primeiro se agrupam vértices sem especificar a ordem pela qual irão ser visitados e depois obtêm-se as rotas para cada veículo tratando os "novos problemas" como um caso de caixeiro viajante.

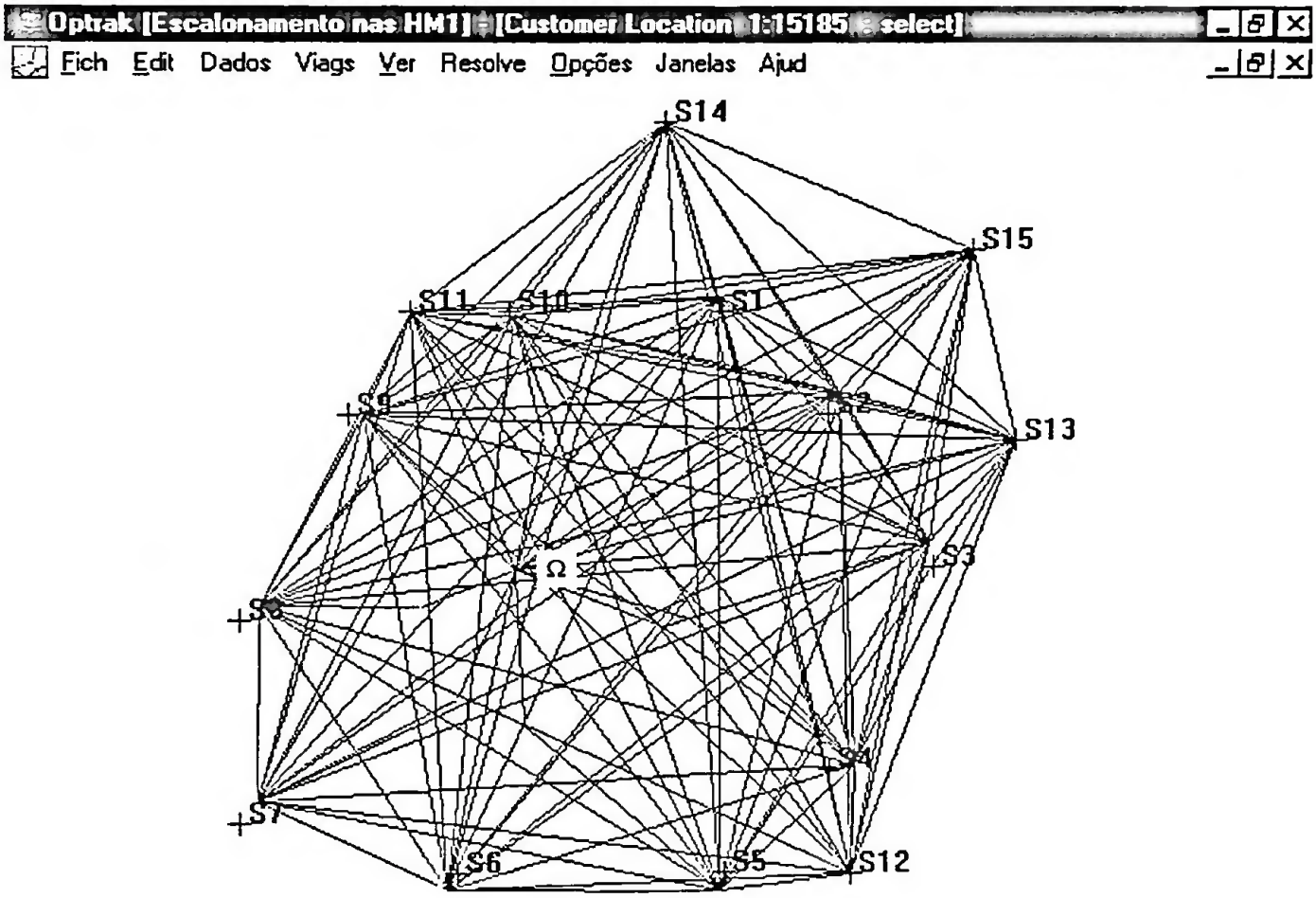
O Optrak opera sobre uma rede digitalizada onde são posicionados os clientes e o depósito, podendo as distâncias entre dois pontos serem medidas com base nas localizações ou serem, como neste caso, arbitradas. Igualmente fixadas, são as velocidades de circulação em cada arco, as taxas de carga e descarga, prioridades das encomendas a entregar nos clientes, o número e capacidade dos veículos e janelas temporais para os clientes receberem as encomendas.

A função objectivo tem coeficientes que ponderam encomendas não satisfeitas, tempo e distância de percurso e veículos usados. Neste caso importa penalizar fortemente encomendas não entregues, correspondendo o resultado final à rota que satisfaça mais encomendas dentro das janelas de tempo respectivas, ou seja, divida mais lotes antes da hora de corte.

7.2.1.Rede

A rede construída para o problema 2.4.2.2. é a que se mostra na figura abaixo:

Figura 6



Os nós correspondem a um ponto central que é o depósito Ω e aos clientes S_1, S_2, ..., S_{15} que representam as saídas dos *OCR*.

O comprimento de um arco em que um dos extremos é o depósito é de 4Km e quando os vértices de um arco são ambos clientes é de 1Km.

7.2.2.Cenários

Com o objectivo de comparação de soluções com o procedimento 4.3., para aferir a qualidade dos resultados produzidos pelo Optrak (aplicado a este caso de escalonamento), *versus* o algoritmo proposto, foram considerados os cenários:

7.2.2.1.Uma máquina, prioridades iguais e tempos de preparação nulos

- Número de veículos: 1;
- Tempos de preparação: como são nulos, a velocidade entre nós foi fixada em 30000Kmh^{-1}, para que as viagens entre clientes sejam efectuadas em tempo "zero";
- Tempos de descarga: dado se admitir uma máquina de divisão, a velocidade de divisão de cada lote é tomada como sendo o dobro da velocidade real, isto é, 1 minuto para cada lote de 1000 cartas;
- Prioridades: todos os lotes têm prioridade 1.

O quadro 7 apresenta os resultados para esta situação:

Quadro 7

Cartas divididos sem atraso por dimensão dos lotes
Uma máquina, sem tempos de preparação e prioridades iguais

Lotes	5000		10000		15000		20000		30000	
Saídas *OCR*	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%
1	120000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8
2	40000	79.7	40000	79.7	30000	59.8	40000	79.7	30000	59.8
3	115000	82.7	110000	79.1	105000	75.5	100000	72.0	90000	64.8
4	35000	52.6	40000	60.1	15000	22.6	40000	60.1	30000	45.1
5	45000	72.1	40000	64.1	45000	72.1	40000	64.1	30000	48.1
6	60000	73.2	60000	73.2	60000	73.2	60000	73.2	60000	73.2
7	80000	74.6	80000	74.6	75000	70.0	80000	74.6	60000	56.0
8	25000	72.1	20000	57.7	15000	43.3	20000	57.7	0	0
9	70000	80.2	70000	80.2	60000	68.8	60000	68.8	60000	68.8
10	20000	66.2	20000	66.2	15000	49.7	20000	66.2	0	0
11	145000	70.2	150000	72.6	105000	50.8	120000	58.1	150000	72.6
12	20000	53.3	30000	80.0	15000	40.0	20000	53.3	0	0
13	65000	37.6	60000	34.7	90000	52.1	120000	69.4	90000	52.1
14	85000	17.4	110000	22.5	135000	27.7	120000	24.6	120000	24.6
15	5000	9.1	0	0	15000	27.3	0	0	30000	54.6
Total	930000	52.2	950000	53.3	900000	50.5	960000	53.9	870000	48.8

7.2.2.2.Duas máquinas, prioridades iguais e tempos de preparação nulos

- Número de veículos: 2;
- Tempos de preparação: como são nulos, a velocidade entre nós foi fixada, igualmente, em 30000Kmh^{-1};
- Tempos de descarga: com duas máquinas, a velocidade de divisão de cada lote é tomada com o verdadeiro valor - 2 minutos para cada lote de 1000 cartas;
- Prioridades: todos os lotes têm prioridade 1.

O quadro 8 mostra os resultados obtidos:

Quadro 8

Cartas divididos sem atraso por dimensão dos lotes
Duas máquinas, sem tempos de preparação e prioridades iguais

Lotes	5000		10000		15000		20000		30000	
Saídas *OCR*	cartas	%	cartas	%	cartas	%	cartas	%	cartas	%
1	120000	78.8	120000	78.8	120000	78.8	120000	78.8	120000	78.8
2	40000	79.7	40000	79.7	30000	59.8	40000	79.7	30000	59.8
3	115000	82.7	110000	79.1	105000	75.6	100000	72.0	90000	64.8
4	40000	60.1	40000	60.1	30000	45.1	40000	60.1	30000	45.1
5	45000	72.1	40000	64.1	45000	72.1	40000	64.1	30000	48.1
6	60000	73.2	60000	73.2	60000	73.2	60000	73.2	60000	73.2
7	80000	74.6	80000	74.6	75000	70.0	80000	74.6	60000	56.0
8	25000	72.1	20000	57.7	15000	43.3	20000	57.7	0	0
9	70000	80.2	70000	80.2	60000	68.8	60000	68.8	60000	68.8
10	20000	66.2	20000	66.2	15000	49.7	20000	66.2	0	0
11	125000	60.5	140000	67.8	150000	72.6	140000	67.8	150000	72.6
12	30000	80	30000	80.0	15000	40.0	20000	53.3	0	0
13	85000	49.2	110000	63.6	105000	60.8	100000	57.9	120000	69.4
14	85000	17.4	80000	16.4	135000	27.7	120000	24.6	240000	49.2
15	15000	27.3	0	0	0	0	0	0	0	0
Total	955000	53.6	960000	53.93	960000	53.9	960000	53.9	990000	55.5

Apesar de *outputs* próximos nestes dois casos não garantirem que o Optrak gere boas soluções para a condição de tempos de preparação não nulos, seria desencorajador se os valores dos quadros 2 e 6 fossem muito díspares dos exibidos nos quadros 7 e 8[30]. Mas o que se observa é que o afastamento entre o Optrak, para a situação de duas máquinas, e o procedimento 4.3. é da ordem dos 10% para os lotes de menor dimensão e aproximando-se do óptimo para os de maior tamanho, "legitimando" o prosseguimento para os cenários seguintes:

[30] A versão que os CTT dispõem apenas permite um máximo de 400 encomendas, não possibilitando fazer lotes de 1000 cartas.

7.2.2.3.Duas máquinas, prioridades diferenciadas e tempos de preparação de 10 minutos

- Número de veículos: 2;
- Tempos de preparação: como se pretende que quando a alimentação da máquina de divisão passe de lotes da saída j para lotes da saída q ($j \neq q$), se tenha um tempo de preparação da máquina de 10 minutos, a velocidade de deslocação entre nós passa a ser de 6 Kmh^{-1}, de modo a que sejam necessários 10 minutos para ir do cliente j para o cliente q.

  Como a distância entre um cliente e o depósito é de 4Km, nenhuma ligação entre clientes se faz pelo depósito. Pelo depósito apenas se passa no início e fim do período de funcionamento do sistema de processamento.
- Tempos de descarga: com duas máquinas, a velocidade de divisão de cada lote é tomada com o verdadeiro valor - 2 minutos para cada lote de 1000 cartas.
- Prioridades: as prioridades de cada lote são as correspondentes às respectivas saídas dos *OCR*, constantes do quadro 1.

7.2.2.4.Duas máquinas, prioridades diferenciadas e tempos de preparação de 5 minutos

- Idêntico ao anterior com a excepção da velocidade entre nós ser de 12 Kmh^{-1}, de modo a que sejam necessários 5 minutos para ir do cliente j para o cliente q.

Face aos valores dos quadros 7 e 8, como $D < 15000$ origina números de lotes em atraso superiores aos restantes, apenas se irão considerar lotes de 15000 e 20000 car-

tas (que asseguram um tempo de ocupação de máquina conveniente). D=30000 embora pareça produzir melhores resultados em termos de lotes tratados, tem uma duração muito longa (um período completo de uma hora).

Os quadros que se seguem expõem os totais obtidos (quadro 9) e os escalonamentos resultantes com tempos de preparação de 5 minutos (quadros 10 e 11):

Quadro 9

Cartas divididas sem atraso por dimensão dos lotes
Duas máquinas, com tempos de preparação e prioridades distintas

Lotes	15000				20000			
Tempos de esvaziamento	5 min		10 min		5 min		10 min	
Saídas *OCR*	cartas	%	cartas	%	cartas	%	cartas	%
1	120000	78.8	120000	78.8	120000	78.8	120000	78.8
2	30000	59.8	30000	59.8	40000	79.7	40000	79.7
3	105000	75.5	105000	75.5	100000	72.0	100000	72.0
4	30000	45.2	30000	45.2	20000	30.1	20000	30.1
5	45000	72.1	45000	72.1	40000	64.1	40000	64.1
6	60000	73.1	45000	54.9	60000	73.2	60000	73.2
7	75000	70.0	45000	44.8	60000	56.0	80000	74.0
8	15000	46.2	15000	46.2	20000	57.7	20000	57.7
9	60000	68.8	60000	68.8	60000	68.8	60000	68.8
10	15000	53.0	15000	53.0	20000	66.2	20000	66.2
11	135000	67.8	135000	67.8	140000	67.8	140000	67.8
12	15000	40.0	15000	40.0	20000	53.3	20000	53.3
13	75000	45.2	60000	34.7	100000	57.9	40000	23.2
14	120000	24.6	120000	24.6	100000	20.5	100000	20.5
15	0	0	0	0	0	0	20000	36.2
Total	900000	50.5	840000	45.5	900000	50.5	880000	49.4

Quadro 10

Duas máquinas, 5 minutos de tempo preparação e prioridades diferenciadas
Escalonamento para lotes de 15000 cartas

HORA	Intervalos em minutos / Saídas *OCR*									
12	0 - 60 —									0 - 60 —
13	0 - 60 S14									0 - 60 —
14	0 - 60 S14					0 - 20 —	20 - 50 S7	50 - 55 —	55 - 60 S13	
15	0 - 60 S14							0 - 55 S13	55 - 60 —	
16	0 - 55 S14	55 - 60 —					0-30 S2	30 - 35 —	35 - 60 S1	
17	0 - 60 S13									0 - 60 S1
18	0 - 5 —	5 - 35 S8	35 - 40 —	40 - 60 S4	0 - 5 S1	5 - 10 —	10 - 40 S12	40 - 45 —	45 - 60 S11	
19	0 - 40 S4	40 - 45 —	45 - 60 S7							0 - 60 S11
20	0 - 60 S7									0 - 60 S11
21	0 - 45 —	45 - 50 S6								0 - 60 S11
22	0 - 60 S6									0 - 60 S11
23	0 - 50 S6	50 - 55 —	55 - 60 S5							0 - 60 S11
24	0 - 60 S5				0 - 15 S11	15 - 20 —	20 - 50 S2	50 - 55 —	55 - 60 S10	
1	0 - 25 S5	25 - 30 —	30 - 60 S13				0 - 25 S10	25 - 30 —	30 - 60 S1	
2	0 - 5 —	5 - 35 S14	35 - 40 —	40 - 60 S9						0 - 60 S1
3	0 - 60 S9									0 - 60 S1
4	0 - 40 S9	40 - 45 —	45 - 60 S3					0 - 5 —	5 - 60 S3	
5	0 - 45 S3	45 - 60 —						0 - 55 S3	55 - 60 —	

Quadro 11

Duas máquinas, 5 minutos de tempo preparação e prioridades diferenciadas
Escalonamento para lotes de 20000 cartas

HORA	Intervalos em minutos / Saídas *OCR*									
12	0 -60 —									0 - 60 —
13	0 - 60 S14								0 -15 —	15 - 60 S14
14	0 - 30 —	30-60 S11								0 - 60 S14
15	0 - 10 S11	10 - 15 —	15 - 55 S7	55-60 —					0 - 55 S14	55 -60 —
16	0 - 40 S6	40 - 45 —	45 - 60 S5							0 - 60 S13
17	0 - 25 S5	25 - 30 —	30 - 60 S7							0 - 60 S13
18	0 -10 S7	10 - 15 —	15 - 55 S6	55 - 60 —					0 - 5 —	5 - 60 S11
19	0 - 40 S12	40 - 45 —	45 - 60 S4							0 - 60 S11
20	0 - 25 S4	25 - 30 —	30 - 60 S8							0 - 60 S11
21	0 - 10 S8	10 - 15 —	15 - 55 S7	55 - 60 —						0 - 60 S11
22	0 - 40 S6	40 - 45 —	45 - 60 S5		0 - 5 S11	5 - 10 —	10 - 50 S10	50 - 55 —		55 - 60 S1
23	0 - 25 S5	25 - 30 —	30 - 60 S10							0 - 60 S1
24	0 - 10 S10	10 - 15 —	15 - 60 S13							0 - 60 S1
1	0 - 35 S13	35 - 40 —	40 - 60 S14							0 - 60 S1
2	0 - 20 S14	20 - 25 —	25 - 60 S9						0 - 55 S1	55 - 60 —
3	0 - 60 S9									0 -60 S2
4	0 - 25 S9	25 - 30 —	30 - 60 S3				0 - 20 S2	20 - 25 —		25 - 60 S3
5	0 - 50 S3	50 - 60 —							0 - 45 S3	45 - 60 —

7.3.Soluções geradas com base num modelo de afectação

Os resultados que se apresentam apenas servem para ilustração do método no caso do CTC de Lisboa já que, como se disse, não existem dados actualizados para os dois outros CTC. A aplicação computacional utilizada foi o TRANSFCL, desenvolvido pela Faculdade de Ciências da Universidade de Lisboa[31].

Apenas se trabalha com lotes de 15000, 20000 e 30000 cartas dado o *software* disponível não permitir lotes de dimensão inferior.

Os valores são idênticos para uma ou duas máquinas, dado que o ganho pela duplicação do número de intervalos usando duas máquinas é anulado pelo tempo de duração de cada intervalo (que passa para o dobro, dado o tempo de processamento de cada tarefa ser multiplicado por 2).

[31] Biblioteca de programas de Investigação Operacional, Departamento de Estatistica e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa.

Quadro 12

Cartas divididas sem atraso por dimensão dos lotes
Sem tempos de preparação e prioridades iguais

Lotes	15000		20000		30000	
Saídas OCR	cartas	%	cartas	%	cartas	%
1	120000	78.8	120000	78.8	120000	78.8
2	30000	59.8	40000	79.7	30000	59.8
3	105000	75.5	100000	72.0	90000	64.8
4	30000	45.1	40000	60.1	30000	45.1
5	30000	48.1	40000	64.1	30000	48.1
6	45000	54.9	60000	73.2	60000	73.2
7	60000	56.0	60000	56.0	60000	56.0
8	15000	43.3	20000	57.7	0	0
9	60000	68.8	40000	45.9	60000	68.8
10	15000	49.7	20000	66.2	0	0
11	120000	58.1	120000	58.1	120000	58.1
12	0	0	20000	53.3	0	0
13	105000	60.8	80000	46.3	120000	69.4
14	270000	55.4	220000	45.1	240000	49.2
15	45000	81.8	40000	72.4	30000	54.6
Total	1050000	58.9	1020000	57.2	990000	55.5

O total de lotes divididos é, dado o método de afectação ser óptimo, coincidente com o procedimento 4.3..

Quadro 13

Uma máquina, sem de tempos de preparação e prioridades iguais
Escalonamento para lotes de 15000 cartas

	Intervalos em minutos / Saídas *OCR*			
HORA				
12	0 - 30 S14			
13	0 - 15 S13	15 - 30 S3	30 - 45 S14	45 - 60 S1
14	0 - 15 S7	15 - 30 S14	30 - 60 S11	
15	0 - 15 S6	15 - 30 S13	30 - 60 S14	
16	0 - 15 S1	15 - 30 S2	30 - 45 S7	45 - 60 S4
17	0 - 15 S5	15 - 30 S6	30 - 45 S9	45 - 60 S11
18	0 - 15 S13	15 - 30 S4	30 - 45 S7	45 - 60 S11
19	0 - 15 S1	15 - 30 S6	30 - 45 S10	45 - 60 S11
20	0 - 15 S1	15 - 30 S9	30 - 45 S11	45 - 60 S13
21	0 - 30 S1	30 - 45 S7	45 - 60 S1	
22	0 - 15 S11	15 - 30 S5	30 - 60 S13	
23	0 - 60 S14			
24	0 - 15 S7	15 - 60 S14		
1	0 - 15 S13	15 - 45 S14	45 - 60 S15	
2	0 - 15 S1	15 - 30 S14	30 - 45 S1	45 - 60 S3
3	0 - 15 S8	15 - 30 S2	30 - 45 S14	45 - 60 S15
4	0 - 15 S15	15 - 45 S9	45 - 60 S3	
5	0 - 45 S3	45 - 60 S14		

O quadro 14 considera a ponderação devida às prioridades, minimizando $\sum_k w_k U_k$.

Embora o complemento ao procedimento 4.3. não garanta o mínimo do número de lotes pesados pelas prioridades, é interessante observar a semelhança dos quadros 6 e 14.

Quadro 14

Cartas divididas sem atraso por dimensão dos lotes
Sem tempos de preparação e prioridades diferenciadas

Lotes	15000		20000		30000	
Saídas OCR	cartas	%	cartas	%	cartas	%
1	120000	78.8	120000	78.8	120000	78.8
2	30000	59.8	40000	79.7	30000	59.8
3	105000	75.5	100000	72.0	90000	64.8
4	30000	45.1	40000	60.1	30000	45.1
5	45000	72.1	40000	64.1	30000	48.1
6	0	0	0	0	0	0
7	75000	70.0	80000	74.6	60000	56.0
8	15000	43.3	20000	57.7	0	0
9	60000	68.8	60000	68.8	60000	68.8
10	15000	49.7	20000	66.2	0	0
11	150000	72.6	140000	67.8	150000	72.6
12	30000	80.0	20000	53.3	0	0
13	135000	78.1	120000	69.4	120000	69.4
14	240000	49.2	220000	45.0	300000	61.5
15	0	0	0	0	0	0
Total	1050000	58.9	1020000	57.2	990000	55.5

CAPÍTULO 8

CONCLUSÕES

Que Caterina não só dispute, mas defina;
não só argumente mas conclua.

Pde. António Vieira, Sermões III

Foi estudado nesta dissertação um problema operacional do subsistema de tratamento automático de correspondências dos CTT - Correios de Portugal, SA, concretamente o escalonamento de lotes de cartas em máquinas de divisão automática de correio, correspondendo estas aos processadores de tarefas.

Para cada tarefa (triagem de um lote de cartas), os *due times*, *ready times*, tempos de execução e prioridades estão definidos, sendo também conhecido o número de tarefas a processar. O objectivo em causa é a minimização do número de lotes divididos com atraso (podendo ou não considerar prioridades diferenciadas para cada lote e tempos de *set-up* das máquinas de divisão dependentes da sequência de lotes), durante o período de funcionamento diário de cada Centro de Tratamento de Correio.

Foi justificado que embora o critério ideal fosse minimizar o número de cartas em atraso e não o número de lotes, as condições do problema inviabilizam o primeiro objectivo.

A solução é proposta em três alternativas, dependendo da satisfação parcial ou integral do enunciado do problema (2.4.2.2.).

O primeiro método de resolução foi desenvolvido admitindo apenas uma máquina, prioridades iguais e unitárias e sem considerar tempos de preparação das divisoras. Ba-

seia-se no algoritmo de Moore-Hodgson alargando-o para este caso de *ready times* não todos nulos, sem ser uma técnica geral para a classe $1 \mid r_k, p_k=1 \mid \sum_k U_k$.

Provou-se que nas condições assumidas é óptimo, o que permite determinar o número mínimo de lotes de cartas que diariamente não é possível tratar em tempo em cada CTC, com uma máquina de divisão. Além deste limite, a forma como o escalonamento é gerado proporciona uma regra para a alimentação das divisoras: os lotes devem ser escolhidos para passarem na máquina divisora por *due time* crescente (que se torna aplicável mesmo na situação de aleatoriedade de tráfego recebido).

Este método, que aparentemente não é satisfatório para o CTC de Lisboa, permite igualmente calcular o mínimo de lotes que é possível deixar em atraso com duas máquinas em paralelo, mantendo as restantes hipóteses. Conseguiu-se este desiderato aplicando o algoritmo desenvolvido para uma máquina com velocidade de processamento supostamente duplicada e provando que num sistema de duas máquinas iguais em paralelo não se conseguem dividir mais lotes que num sistema de uma máquina com velocidade duas vezes superior.

Mostrou-se também como a partir de um escalonamento de lotes de cartas numa máquina, se pode conceber um escalonamento para duas máquinas em paralelo com velocidade 50% menor, mantendo o número de cartas divididas sem atraso e, desta forma, resolver igualmente o problema para o CTC de Lisboa.

Existindo prioridades diferenciadas, embora não se garanta a minimização de $\sum_k w_k U_k$, o algoritmo pode, opcionalmente, ser complementado de forma a, conservando o número óptimo de lotes divididos sem atraso, lotes mais prioritários não dei-

xem de ser tratados para o serem lotes de prioridade inferior (sempre que tal seja possível). Neste caso a regra do *due time* crescente não é inteiramente utilizável, mas pode pensar-se em respeitá-la a menos que exista um número elevado de lotes de prioridade superior para dividir.

Uma questão despertada na modelização do problema foi a dimensão dos lotes: não muito pequenos para não se provocarem muitas mudanças de programa nas divisoras, nem muitos grandes para não inviabilizar a divisão de quantidades significativas de cartas.

Para os valores ensaiados, observa-se que com lotes de 1000 ou 5000 se conseguem processar mais cartas do que para dimensões superiores.

Igualmente tomando lotes de 1000 e 5000 e considerando prioridades diferenciadas, para as saídas 3 e 9, mais prioritárias que 14, obtêm-se percentagens de cartas processadas em tempo maiores, do que com lotes de 15000 ou 30000 cartas. No entanto, fixando os lotes em 1000 ou 5000 para ganhar em quantidade total de cartas divididas, causam-se frequentes mudanças de programa nas divisoras, sobretudo entre as 12 e as 22 horas, o que é dificilmente aceitável na perspectiva operacional.

A opção por dimensões de 10000, 15000 ou 20000 cartas, resulta em decréscimos inferiores, respectivamente, a 1%, 3% e 6%, mas com menos quebras na sequência de divisão.

No CTC de Lisboa, seguindo a técnica de fraccionar os lotes ao meio e distribui-los por cada uma das divisoras, a alternativa mais aconselhável é a de 20000 cartas/lote (que não prejudica com significado as saídas mais prioritárias e garante um tempo mínimo de 20 minutos em cada máquina), excepto para os primeiros períodos de funcionamento onde se podem fazer lotes de 10000 cartas para não ter as divisoras inactivas.

Nestes períodos, afastados das horas de corte para qualquer lote e sem correr o risco de aumentar o número de "atrasados", pode considerar-se uma variante: apenas "partir" os lotes em fracções de 10000 cartas se dois ou mais lotes da mesma saída estão contiguamente sequenciados num período, colocando todos os lotes de 10000 cartas alternadamente em cada uma das duas máquinas divisoras.

Obtém-se, assim, um máximo de cartas divididas em tempo de cerca de 60% do tráfego total[32].

A segunda abordagem, considerando o problema na versão completa, foi suportada na modelização deste caso como um *vehicle routing problem*, utilizando um *software* comercial - Optrak - para as tarefas computacionais.

A avaliação do comportamento do Optrak para a condição de prioridades iguais e sem tempos de preparação num sistema de duas máquinas em paralelo, comparando-o com os alcançados com o algoritmo desenvolvido para esta situação particular, desviaram-se deste em menos de 10%. Este bom desempenho indicia que, com tempos de 5 ou 10 minutos para mudar programas nas divisoras, não se pode esperar dividir muito mais que 54% das cartas no CTC de Lisboa[33]. Evidentemente que a redução daquele tempo é um factor a ter em conta para melhorar o rendimento da divisão.

Quanto à dimensão, 15000 cartas/lote é um número que proporciona resultados equilibrados em tempo de ocupação mínimo das divisoras e lotes divididos.

---

[32] Apenas com 95% do correio azul existe a obrigatoriedade de entrega no dia útil seguinte ao da recepção e alguns lotes podem ser ainda tratados manualmente.

[33] Um bom utilizador do Optrak pode parametrizá-lo para melhor adequação ao caso presente de modo a incrementar, ainda que ligeiramente, os resultados obtidos.

Se a variação diária do número de lotes em cada período for apreciável, o estabelecimento de uma regra de sequência torna-se aqui mais difícil, mas o escalonamento obtido pode ser a primeira aproximação para cada dia de operação.

O terceiro método apresentado, tratando o escalonamento como um problema de afectação, embora manifeste os inconvenientes apontados para gerar escalonamentos em cada CTC (embora para lotes acima das 10000 cartas como não há muitas combinações de lotes possíveis de tratar em cada período essas desvantagens se "diluam") , revela interesse (para tempos de preparação nulos), quando se procura determinar se a transferência de correio inter-CTC permite diminuir a quantidade de lotes em atraso, quando se consideram todas as divisoras num sistema integrado.

Duas ordens de razões explicam o facto de apenas se ter avançado o aspecto conceptual: dificuldade em encontrar aplicações em micro-computador para as dimensões do problema e indisponibilidade de dados para os CTC de Coimbra e Porto.

No entanto, é uma questão formalmente resolvida e cuja aplicação pode ser realizada recorrendo a computadores de maior porte (programando o algoritmo da afectação ou procurando *software* já desenvolvido) e estimando o diagrama de cargas para aqueles dois CTC.

Como último apontamento, importa realçar que este trabalho também pode ter aplicabilidade para ensaiar alternativas de agrupamentos de destinos nas saídas dos *OCR*, aos reflexos provocados por alterações às horas de corte para cada destino e aos momentos de chegada de correio aos CTC.

"- Há que ler, há que ler ..."

Eça de Queiroz, A cidade e as serras

[1] Pilar, Norberto (1996), "Os Correios no mundo de hoje", Cadernos de Economia, Nº 36, Jul/Set 1996, pp 48-52.

[2] Rosa, Albano (1990), Estudo sistémico da substituição de um centro de tratamento de correio e sua rede de transporte, tese de mestrado, IST,UTL.

[3] Cardoso, Eurico (1984), Os Correios, os selos e a filatelia, Lisboa: edição de autor.

[4] CTT (1996), Correios de Portugal, Relatório e contas do ano de 1995, Lisboa.

[5] Tavares, L. V., Oliveira R., Hall Themido, Isabel e Correia, F. Nunes (1996), Investigação Operacional, Lisboa: Mc Graw-Hill de Portugal.

[6] Morton, T. e Pentico, D. (1993), Heuristic Scheduling Systems With applications to production and project management, New York: Wiley.

[7] Ramalhete, M., Guerreiro, J., e Magalhães, A.(1984), Programação Linear, vol I, Lisboa: McGraw-Hill.

[8] Ramalhete, M., Guerreiro, J., e Magalhães, A.(1985), Programação Linear, vol II, Lisboa: McGraw-Hill.

[9] Ackoff, R., e Sasieni, M. (1968), Fundamentals of Operations Research, New York:Wiley.

[10] French, S. (1990), Sequence and scheduling, an introduction to the mathematics of job-shop, London: Ellis Horwoord.

[11] Lawler, E., Lenstra, J., Rinnooy Kan, A. e Shmoys D. (1993), Sequencing and complexity, in: S. C. Graves et al (ed), Handbooks in operational research and management science, Amsterdam: Elsevier.

[12] Centeno, M. (1993), Métodos heurísticos para o problema de escalonamento de tarefas numa máquina com custo de antecipação e atraso, tese de mestrado, ISEG,UTL.

[13] Kise, H. , Ibaraki, T. e Mine, H. (1978), "A solvable case of the one-machine scheduling problem with ready and due times", Operations research,Vol. 26, N°1, pp 121-126.

[14] Madureira, A.e Sousa, J. (1996), "Aplicações de meta-heurísticas a problemas de escalonamento de uma única máquina", Investigação Operacional, Vol 16, N° 2, pp 115-133.

[15] Brucker, P., Scheduling algorithms (1995), Berlin: Springer-Verlag.

[16] Eilon, S., Watson-Gandy, C.D.T., Christofides, N., Distribution Management: Mathematical modelling and practical analysis (1971), London: Griffin.

[17] Notas sobre o Optrak 3.1., Optimiza, Lda, Lisboa (1995).

[18] Stewart, I., Os problemas da matemática (1996), Lisboa: Gradiva.

ANEXOS

"- Quereis dizer que não há mais nada
que vós tenhais poder para me dizer? "

Umberto Eco, O nome da rosa

Quadro A1

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 5000 cartas

HORA	Intervalos em minutos / Saídas *OCR*								
12	0 - 10 S11	10 -15 S6	15 - 20 S7	20 - 30 S13	30 - 35 S1	35 - 60 S14			
13	0 - 5 S4	5 - 15 S11	15 - 20 S7	20 - 25 S13	25 - 35 S1	35 - 40 S5	40 - 60 S14		
14	0 - 10 S11	10 - 15 S6	15 - 20 S7	20 - 30 S13	30 - 35 S12	35 - 40 S1	40 - 45 S8	45 - 60 S14	
15	0 - 5 S4	5 - 15 S11	15 - 20 S6	20 - 25 S10	25 - 30 S7	30 - 35 S13	35 - 40 S1	40 - 45 S5	45 - 60 S14
16	0 - 10 S4	10 - 40 S11	40 - 50 S6	50 - 55 S10	55 - 60 S7				
17	0 - 5 S11	5 - 10 S6	10 - 25 S7	25 - 60 S13					
18	0 - 10 S4	10 - 35 S11	35 - 45 S6	45 - 55 S7	55 - 60 S13				
19	0 - 5 S4	5 - 15 S11	15 - 20 S6	20 - 25 S10	25 - 30 S7	30 - 55 S13	55 - 60 S12		
20	0 - 5 S4	5 - 25 S11	25 - 30 S6	30 - 40 S7	40 - 55 S13	55 - 60 S12			
21	0 - 15 S11	15 - 20 S6	20 - 25 S10	25 - 35 S7	35 - 45 S13	45 - 55 S12	45 - 60 S1		
22	0 - 10 S11	10 - 15 S6	15 - 20 S7	20 - 30 S13	30 - 60 S1				
23	0 - 55 S1	55 - 60 S5							
24	0 - 30 S5	30 - 50 S8	50 - 60 S14						
1	0 - 5 S13	5 - 60 S14							
2	0 - 5 S12	5 - 10 S1	10 - 15 S5	15 - 60 S14					
3	0 - 60 S14								
4	0 - 15 S2	15 - 45 S9	45 - 60 S3						
5	0 - 60 S3								

Quadro A2

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 15000 cartas

HORA	Intervalos em minutos / Saídas *OCR*			
12	0 - 30 S14			
13	0 - 15 S11	15 - 30 S13	30 - 45 S1	45 - 60 S14
14	0 - 15 S11	15 - 30 S7	30 - 45 S14	45 - 60 S3
15	0 - 15 S6	15 - 30 S13	30 - 60 S14	
16	0 - 15 S4	15 - 45 S11	45 - 60 S7	
17	0 - 15 S11	15 - 30 S6	30 - 60 S13	
18	0 - 15 S4	15 - 30 S11	30 - 45 S7	45 - 60 S13
19	0 - 15 S11	15 - 30 S6	30 - 45 S10	45 - 60 S13
20	0 - 15 S7	15 - 30 S7	30 - 45 S13	45 - 60 S12
21	0 - 15 S11	15 - 30 S7	30 - 45 S13	45 - 60 S1
22	0 - 15 S11	15 - 30 S6	30 - 60 S1	
23	0 - 45 S1	45 - 60 S5		
24	0 - 30 S5	30 - 45 S8	45 - 60 S14	
1	0 - 15 S13	15 - 60 S14		
2	0 - 15 S12	15 - 30 S1	30 - 60 S14	
3	0 - 60 S14			
4	0 - 15 S2	15 - 45 S9	45 - 60 S3	
5	0 - 60 S3			

Quadro A3

Uma máquina, sem tempos de preparação e prioridades iguais
Escalonamento para lotes de 30000 cartas

	Intervalos em minutos / Saídas *OCR*	
HORA		
12	0 - 30 —	30 - 60 S14
13	0 -60 —	
14	0 - 30 S11	30 - 60 S14
15	0 - 30 S13	30 - 60 S14
16	0 - 30 S11	30 - 60 S7
17	0 - 30 S6	30 - 60 S13
18	0 - 30 S4	30 - 60 S11
19	0 - 30 S13	30 - 60 S1
20	0 - 30 S11	30 - 60 S7
21	0 - 30 S13	30 - 60 S1
22	0 - 30 S11	30 - 60 S6
23	0 - 30 S1	30 - 60 S5
24	0 - 60 S14	
1	0 - 60 S14	
2	0 - 30 S1	30 - 60 S14
3	0 - 60 S14	
4	0 - 30 S9	30 - 60 S3
5	0 - 60 S3	

Quadro A4

Duas máquinas, 10 minutos de tempo preparação e prioridades diferenciadas
Escalonamento para lotes de 15000 cartas

HORA	Intervalos em minutos / Saídas *OCR*									
12	0 - 60 —									0 - 60 —
13	0 - 15 —	15 - 45 S13	45 - 55 —	55 - 60 S11						0 - 60 S14
14	0 - 25 S11	25 - 35 —	35 - 60 S7							0 - 60 S14
15	0 - 5 S7	5 - 15 —	15 - 45 S9	45 - 55 —	55 - 60 S6			0 - 30 S14	30 - 40 —	40 - 60 S13
16	0 - 25 S6	25 - 35 —	35 - 60 S1					0-40 S13	40 - 50 —	50 - 60 S7
17	0 - 60 S1							0 - 20 S7	20 - 30 —	30 - 60 S6
18	0 - 35 S1	35 - 45 —	45 - 60 S12						0 - 10 —	10 - 60 S5
19	0 - 15 S12	15 - 25 —	25 - 60 S4					0 - 10 S5	10 - 20 —	20 - 60 S11
20	0 - 25 S4	25 - 35 —	35 - 60 S3							0 - 60 S11
21	0 - 60 S3									0 - 60 S11
22	0 - 35 S3	35 - 45 —	45 - 60 S2							0 - 60 S11
23	0 - 45 S2	45 - 55 —	55 - 60 S10					0 - 20 S11	20 - 30 —	30 - 60 S6
24	0 - 25 S10	25 - 60 —					0 - 10 —	10 - 40 S7	40 - 50 —	50 - 60 S5
1	0 - 60 S1							0 - 20 S5	20 - 30 —	30 - 60 S13
2	0 - 60 S1								0 - 10 —	10 - 60 S14
3	0 - 10 —	10 - 40 S8	40 - 50 —	50 - 60 S9				0 - 40 S14	40 - 50 —	50 - 60 S9
4	0 - 20 S9	20 - 30 —	30 - 60 S3						0 - 50 S9	50 - 60 —
5	0 - 60 S3									0 - 60 —

Quadro A5

Duas máquinas, 10 minutos de tempo preparação e prioridades diferenciadas
Escalonamento para lotes de 20000 cartas

	Intervalos em minutos / Saídas *OCR*								
HORA									
12	0 -60 —								0 - 60 —
13	0 - 60 S14							0 - 20 —	20 - 60 S11
14	0-60 S11						0 - 10 —	10 - 50 S13	50 - 60 —
15	0 - 40 S11	40 - 50 —	50 - 60 S11				0 - 40 S7	40 - 50 —	50 - 60 S1
16	0 - 40 S6	40 - 45 —	45 - 60 S5						0 - 60 S1
17	0 - 10 S11	10 - 20 —	20 - 60 S3				0 - 10 S1	10 - 20 —	20 - 60 S4
18	0 -10 —	10 - 60 S9					0 - 10 —	10 - 50 S11	50 - 60 —
19	0 - 30 S9	30 - 40 —	40 - 60 S7				0 - 40 S12	40 - 50 —	50 - 60 S11
20	0 - 60 S7								0 - 60 S11
21	0 - 10 —	10 - 50 S10	50 - 60 —						0 - 60 S11
22	0 - 60 S6						0 - 30 S11	30 - 40 —	40 - 60 S7
23	0 - 60 S5						0 - 20 S7	20 - 30 —	30 - 60 S12
24	0 - 10 —	10 - 50 S5	50 - 60 S5				0 - 10 S12	10 - 20 —	20 - 60 S1
1	0 - 30 S5	30 - 40 —	40 - 60 S9						0 - 60 S1
2	0 - 20 S9	20 - 30 —	30 - 60 S8						0 - 60 S1
3	0 - 10 S8	10 - 20 —	20 - 60 S2				0 - 10 —	10 - 50 S14	50 - 60 —
4	0 - 10 —	10 - 50 S15	50 - 60 —						0 - 60 S3
5	0 - 40 S3	40 - 60 —							0 - 60 S3