OPTIMAL SCHEDULING OF MULTIPURPOSE BATCH/SEMI-CONTINUOUS PLANTS

by
Lazaros Papageorgiou

July 1994

A thesis submitted for the degree of Doctor of Philosophy of the University of London and for the Diploma of Membership of the Imperial College

Department of Chemical Engineering and Chemical Technology
Imperial College of Science, Technology and Medicine
London SW7 2BY, U.K.

To my parents

Abstract

The batch and semi-continuous modes of operation are suitable for plants manufacturing relatively large numbers of low-volume, high-value added products (*e.g.* pharmaceuticals) which compete for scarce common resources over a given time horizon. The last decade has seen a renewed interest in the batch mode of operation in the chemical industry particularly because of the commercial environment which has become more uncertain, complex and competitive.

This work considers two distinct problems, namely those of campaign planning/scheduling and energy integration in multipurpose plant operation.

In general, the planning problem is relevant to long time horizons and when reliable product demands are available. Then, it is preferable to operate the plant in a regular fashion, splitting the whole horizon into a number of long periods of time ("campaigns"), each dedicating plant resources to the manufacture of a single product or a subset of the potential products. This simplifies the management and the control of the plant, and minimises change-overs between various products. The processing recipes are of arbitrary complexity including batch mixing, splitting, material recycles, shared intermediates *etc.*, whilst the process resources are characterised by general, multipurpose functionality. The multiple campaign planning problem involves the determination of the campaign structure (*i.e.* number, duration and constituent products), as well as the details of resource utilisation for every campaign. A cyclic operating schedule, repeated at a fixed frequency within each campaign, must also be determined. The problem is formulated as a mixed integer linear programming (MILP) model, and is solved via a branch-and-bound technique.

The optimal exploitation of energy integration in the operation of multipurpose batch/semi-continuous plants is also considered. The overall problem of energy integration in batch processes is decoupled hierarchically into two subproblems. The first subproblem involves determining the optimal operating policy for two or more energy coupled processing steps. This is formulated as a multistage optimal control problem. The second subproblem, formulated as a MILP/MINLP model, involves the determination of an optimal operating schedule which may involve both the original (stand-alone) processing steps and the energy-coupled steps determined by the first subproblem. The two potential modes of energy integration, direct and indirect, are taken rigorously into account within the framework of a general production scheduling formulation for multipurpose plants.

Acknowledgements

I would like to thank Dr. Costas C. Pantelides for his excellent supervision, co-operation and motivation throughout this project.

I appreciate the recommendations and many helpful discussions with Dr. Nilay Shah.

I am grateful to Matheos Charalambides for his help in constructing and implementing the dynamic batch reaction and distillation models used for the work presented in chapter 7 of this thesis.

I would like to ackowledge the invaluable support of my family during my studies.

Finally, many thanks to all my friends and colleagues for providing me a pleasant stay in London.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Although the continuous mode of operation is the dominant mode of manufacture for products produced in large quantities (*e.g.* commodity chemicals), the alternative batch and semi-continuous modes of operation are suitable for plants manufacturing relatively large numbers of low-volume, high-value added products (*e.g.* pharmaceuticals) which compete for scarce common resources over a given horizon of interest.

The last decade has seen a renewed interest in the batch mode of operation in the chemical industry (Parakrama, 1985; Hasebe and Hashimoto, 1992), particularly because of the commercial environment which has become more uncertain, complex and competitive. Typically, the need for product customisation and diversification has led to a larger number of end-products being produced in common manufacturing facilities. Processing recipes for the manufacture of various products as well as modern equipment and resources in general are becoming more flexible and complex, thus complicating plant management and control. Furthermore, the uncertainty with respect to sudden changes of current outstanding orders, market economics and resource availabilities requires rapid responses.

The above factors lead to the need to allocate shared resources to competing demands over time. This gives rise to difficult scheduling problems of high complexity and dimensionality whose efficient solution requires that a systematic approach be adopted.

The basic benefits of optimal scheduling include:

- increased productivity;
- improved resource utilisation;
- enhanced business responsiveness;
- increased customer satisfaction;
- identification of possible production bottlenecks;
- reduced costs (*e.g.* change-over, inventory); and
- improved plant reliability and availability.

Traditionally, the general process scheduling problem has been considered to comprise two subproblems, sequencing and scheduling. In the former problem, the sequence of batches of products is determined so as to fulfil outstanding orders, while in the latter, the exact timings of the unit operations are obtained in order to achieve the resulting sequence of batches from the first step. Although the above hierarchical decomposition is straightforward for simple processing networks (*e.g.* serial processing networks without batch splitting or mixing), there is no clear distinction between the two for more complex plants where batches of various products may be split or mixed, products may share common intermediates *etc.* The two subproblems must therefore be considered simultaneously.

Usually, batch plants operate with considerable flexibility, accommodating large numbers of products using the same resources. Depending on the production routes, batch plants are distinguished into multiproduct and multipurpose ones. In the former case, all the products are manufactured following the same production route whilst in the latter, more general case, products may follow different routes through the plant.

Depending on the business environment as well as on the availability of accurate information concerning future demands for all potential products being produced, batch/semi-continuous plants usually operate in one of two distinct modes: short-term and campaign mode. In the absence of reliable, long-term information, plants tend to be operated in a *short-term mode*, with the production plan being largely determined by the currently outstanding orders. This usually implies relatively short planning horizons (typically ranging from one to a few weeks). The pattern of operations in any schedule is usually irregular with production being different from one planning interval to the next due to the varying nature of outstanding orders. For example, plants based on production to specific customer requirements (*e.g.* paints, lubricants) may be operated in this mode.

If, on the other hand, reliable long-term demand predictions are available, it is often preferable to partition the planning horizon into a number of relatively long periods of time ("campaigns"), each dedicated to the production of a single product or a subset of products. This *campaign mode* operation may result in important benefits such as minimising the number and costs of change-overs when switching the production from one product to another. The complexity of management and control of the plant operation is further reduced by operating the plant in a more regular fashion, such as in a cyclic mode within each campaign, with the same pattern of operations being repeated at a constant frequency. Typical campaign lengths are from a few weeks to several months, with cycle times ranging from a few hours to a few days. The campaign mode of operation is often used for the manufacture of "generic" materials (*e.g.* base pharmaceuticals) which are produced in relatively large amounts and are then used as feedstocks for downstream processes producing various more specialised final products.

1.1 Thesis Outline

The short-term scheduling problem for processes described by general recipes and plants involving flexible production resources has been considered in detail by Kondili (1988) and Shah (1992). This thesis is concerned with campaign planning and scheduling for equally general processes and plants. Also, interactions between energy integration and optimal scheduling and unit operation are studied. All these features have been incorporated within a single mathematical framework which is described.

This thesis comprises two main parts. The first deals with the problem of campaign planning/scheduling of multipurpose batch/semi-continuous plants, and consists of the following four chapters: Chapter 2 presents a general overview of the problem, covering the characteristics of batch processes and providing a definition of the campaign planning/scheduling problem. A single-level MILP model is described in chapter 3 and its efficient solution is discussed in chapters 4 and 5 by applying simultaneous and decomposition schemes respectively.

The second part of the thesis, involving chapters 6 and 7, tackles the problem of energy integration in multipurpose plants. In chapter 6, the optimal scheduling of energy integrated plants is considered using a systematic mathematical framework taking into account stand-alone and energy-integrated operations.

Chapter 7 describes techniques for determining optimal operating policies for energy-coupled unit operations through optimal control analysis.

Finally, an overview of the thesis is given in chapter 8, which also outlines possible future directions related to this work.

PART I

CAMPAIGN PLANNING/SCHEDULING FOR MULTIPURPOSE BATCH/SEMI-CONTINUOUS PLANTS

Chapter 2

Campaign Planning/Scheduling — Background

The next four chapters deal with the campaign planning/scheduling problem for batch/semi-continuous plants. In this chapter, the general characteristics of batch/semi-continuous plants are described, the overall problem of campaign planning is defined, and a review of the relevant literature on campaign planning is presented.

2.1 Representation of Batch/Semi-continuous Processes

One key feature of the approach used in this thesis is the use of the *State-Task Network* (STN) process representation (figure2.1) introduced by Kondili *et al.* (1988, 1993).

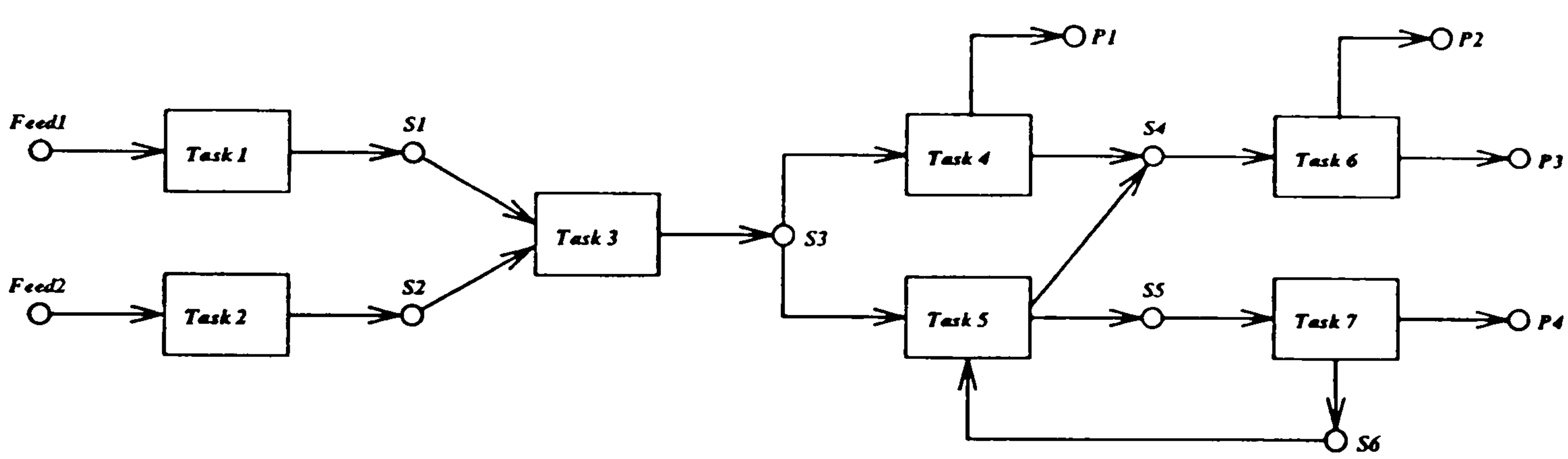


Figure 2.1: State-Task Network Process Representation

STNs are directed graphs, comprising two types of nodes, namely state and task nodes. State nodes (denoted by circles) correspond to feeds, intermediates and final products, whilst task nodes (denoted by rectangles) represent physico-chemical operations transforming material from one type (task feeds) to another (task products).

In contrast to conventional flowsheets, the STN representation describes batch/semi-coninuous recipes unambiguously, thus accommodating complex processes with material recycles, shared intermediates, multiple processing routes to the same type of material as well as mixing and splitting of individual batches.

The STN framework makes the following simplifying assumptions:

- the no pre-emptive mode of operation is enforced (*i.e.* no task may be interrupted once it has started);

- the task processing times are fixed and known *a priori*. Different output states of the same task may have different processing times;
- the material is transferred instantaneously from states to tasks and vice versa; and
- all data are deterministic and fixed over the planning horizon of interest.

2.2 Resource Characterisation

The STN representation is concerned *only* with process recipes and does not involve any information associated with the plant resources. In this context, resources include processing and storage equipment, as well as the available utilities. In general, all available resources are flexible and multipurpose. Their characteristics are outlined below.

Processing equipment is suitable for performing various processing tasks. Each item is characterised by its nominal capacity, but may have different utilisation factors for different tasks due to safety and/or other operating bounds. For example, a mixer requires a minimum batchsize to ensure that its impeller be covered, and a reactor with a heating or cooling coil will have operating bounds on the volume of its contents in order to ensure effective temperature control. Apart from performing tasks, each item of equipment may also be suitable for storing various states of material.

Although there may often be no clear distinction between processing and storage equipment, the latter usually refers to items that are suitable only for storing processing states over certain periods of time. Based on the way the storage resources are used for storing various states, we can distinguish three storage modes, namely *dedicated*, *multipurpose* and *shared* storage modes. The first is applicable when a storage tank is suitable for storing exactly one state. The multipurpose mode refers to cases where a storage tank can store more than one processing state but at most one at any given time. The shared storage mode refers to the possibility of storing various products within the same storage resource (*e.g.* warehouses) where more than one processing state can share the same facility at the *same* time as long as their cumulative level does not exceed the overall capacity of the resource.

Finally, the available utilities of the plant (*e.g* steam, cooling water, manpower *etc.*) may have multipurpose functionality, being suitable for more than one processing task. The amount of a utility consumed by a task may be constant, or a function of the batchsize. Also, the availability and cost of the utility may vary with time.

2.3 Campaign Operation for Multipurpose Plants

Campaign operation is more appropriate for plants operating under stable demand patterns over long planning horizons. The basic advantages of campaign operation are lower inventory costs, fewer change-overs and improved plant operability.

2.3.1 Motivation for Campaign Operation

In order to illustrate why the campaign mode of operation is necessary or desirable for certain cases, we consider a simple multiproduct plant manufacturing three products, *Prod1*, *Prod2* and *Prod3*, from three feedstocks, *Feed1*, *Feed2* and *Feed3*, respectively. All the products are made via the following processing steps:

[1] *Reaction*, a batch task which transforms feedstocks to intermediates after a processing time of 2 hours.

[2] *Filtering*, a batch task that starts immediately after the reaction task, resulting in final products from intermediates after 1 hour.

The corresponding STN is illustrated in figure 2.2.

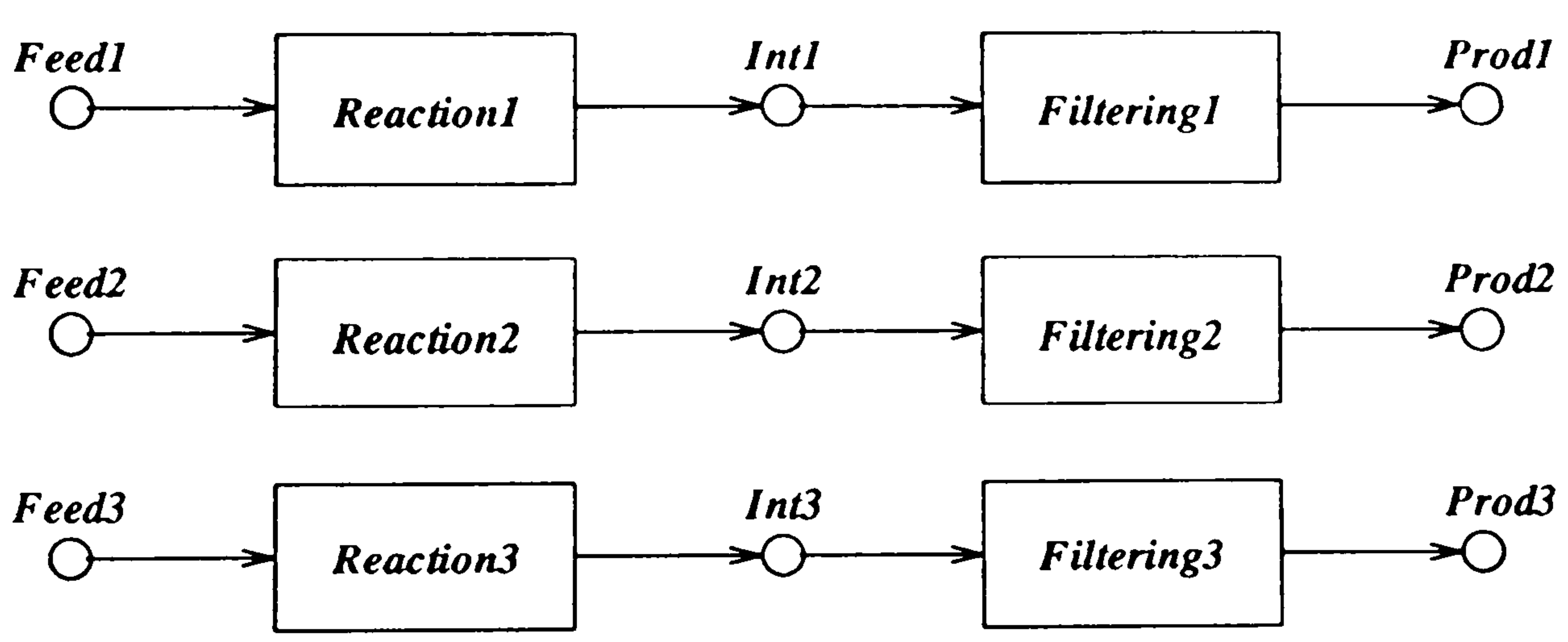


Figure 2.2: STN for 3-product Plant

Minimum production requirements of 400, 400 and 800 tonnes are imposed on *Prod1*, *Prod2* and *Prod3*, respectively at the end of a 3-month planning horizon. Unlimited market demand exists for all the products, each of which is assumed to have a unit price.

The plant processing equipment comprises a single 5-tonne reactor suitable for performing all reaction tasks, and two 3-tonne filters for performing all filtering tasks. A 4-hour cleaning task is required when switching production on the reactor. However, no cleaning is necessary for the filters.

There is no storage capacity for intermediates, and the plant operates in "zero-wait" mode. Also, unlimited storage is assumed for all feedstocks and final products.

We assume that the plant will operate in a cyclic mode with a 24 hour cycle time. A production plan along with a detailed operating schedule are required to maximise the production amounts subject to fulfilling the minimum production requirements.

First, we derive a periodic schedule for a single campaign spanning the whole planning horizon, with all three products being manufactured in each cycle. The optimal operating schedule is shown in figure 2.3.

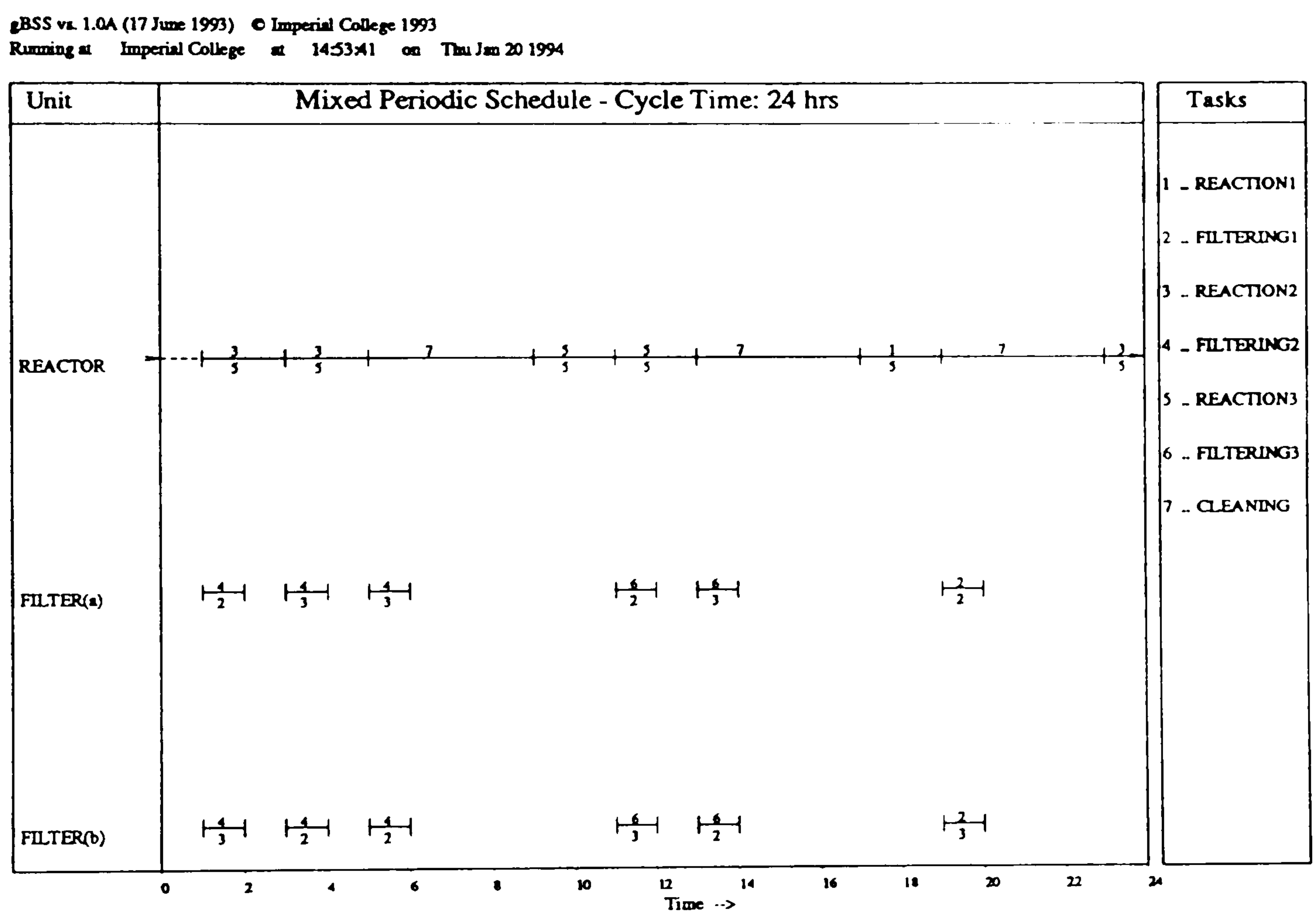


Figure 2.3: Single-campaign Gantt Chart

The above Gantt chart represents unit-task allocations as horizontal solid line segments. The number above each segment denotes the task (see key in right hand side of figure 2.3) while the number below each segment denotes the corresponding batchsize. Note that task 7 denotes the reactor cleaning operation.

Due to the cyclic nature of the schedule, some processing tasks may span successive cycles, starting in the previous cycle and extending across the boundary into the next one (the so called "wrap-around" effect). This is shown as a dashed line in figure 2.3.

The above production plan leads to an objective function of 2667 relative cost units (*rcu*). It can be seen from figure 2.3 that three cleaning tasks are required per cycle in order to switch production between different products.

Due to the relatively large durations of the cleaning operations, a multiple campaign production plan might be more appropriate. More specifically, we consider three campaigns where each manufactures only one of the above products. Furthermore, we assume that when switching the production from one campaign to another, an inter-campaign shut-down of one day is required due to maintenance and plant reconfiguration. In this case, the final production plan involves three campaigns with durations 768, 576 and 768 hours respectively, while two days are spent for inter-campaign change-overs as shown in figure 2.4.

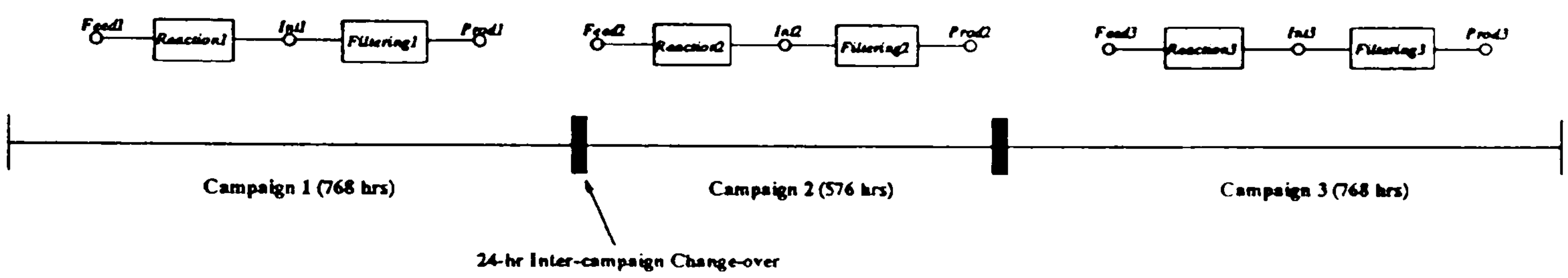


Figure 2.4: 3-campaign Production Plan

By operating the plant in multiple campaign fashion, the objective function increases to the value of 5213 *rcu*, leading to an improvement of 95% due to the elimination of cleaning times within each campaign, leaving only a relatively short downtime of two days duration between successive campaigns. The improved resource utilisation for the first campaign manufacturing *Prod1* is illustrated in figure 2.5. Similar schedules are derived for the other two campaigns.

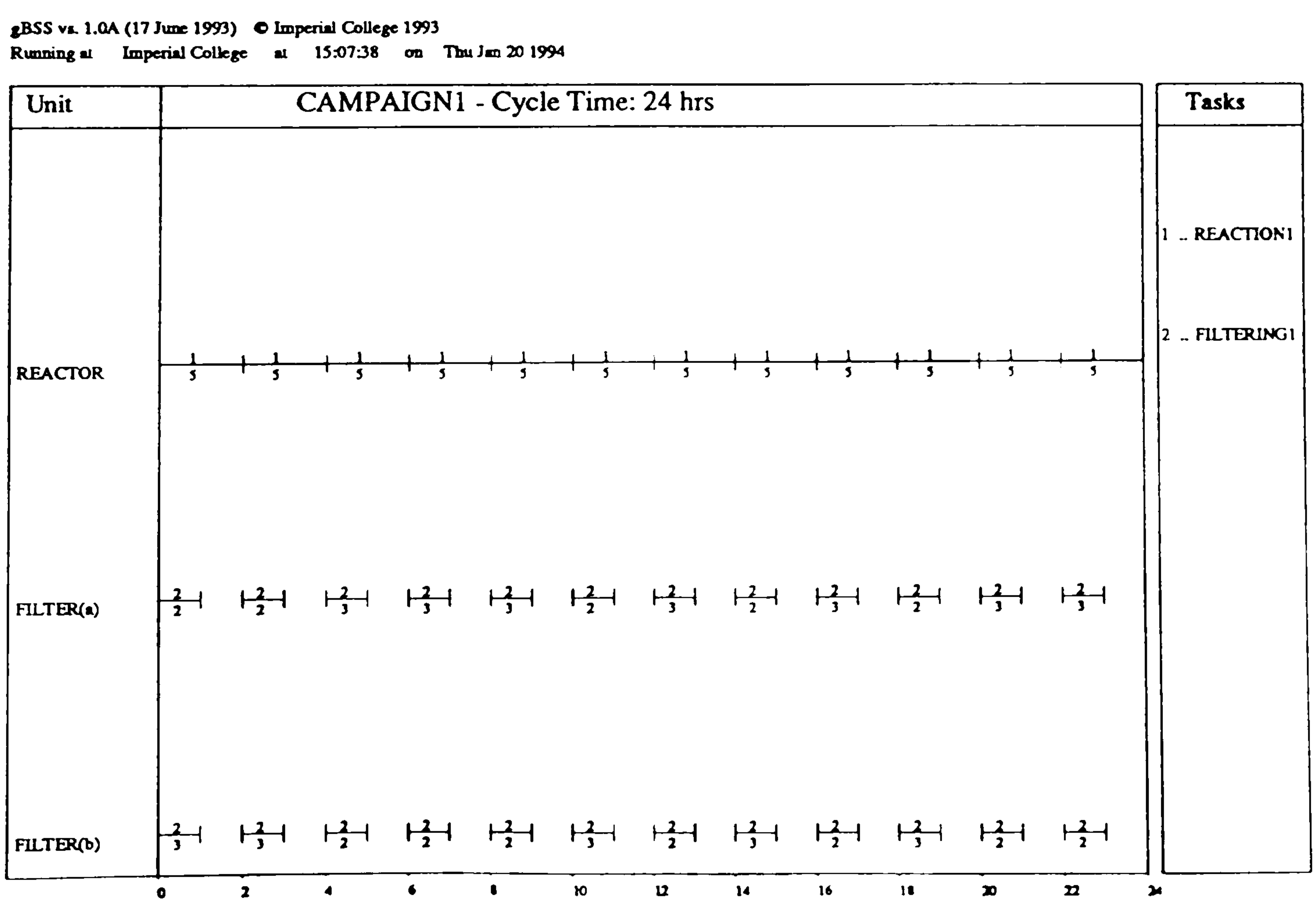


Figure 2.5: Gantt Chart for Campaign 1

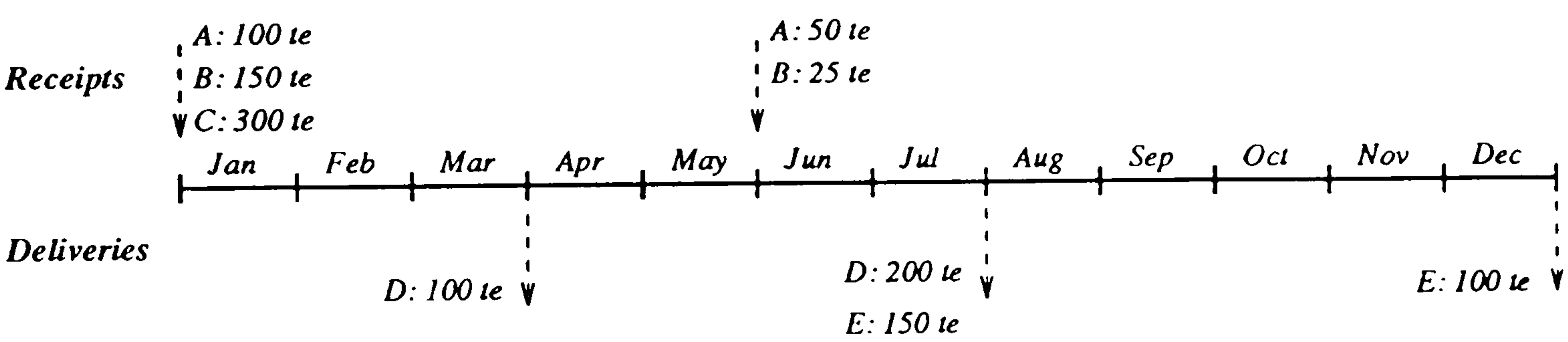
2.3.2 Product Demands, Raw Material Supplies and Inventories

An important aspect of the operation of any plant is its interaction with the material flows from suppliers and to customers. Deliveries and receipts of different types of materials (raw materials, intermediates and products) may be point events, taking place at the end of, or at distinct points during the time horizon.

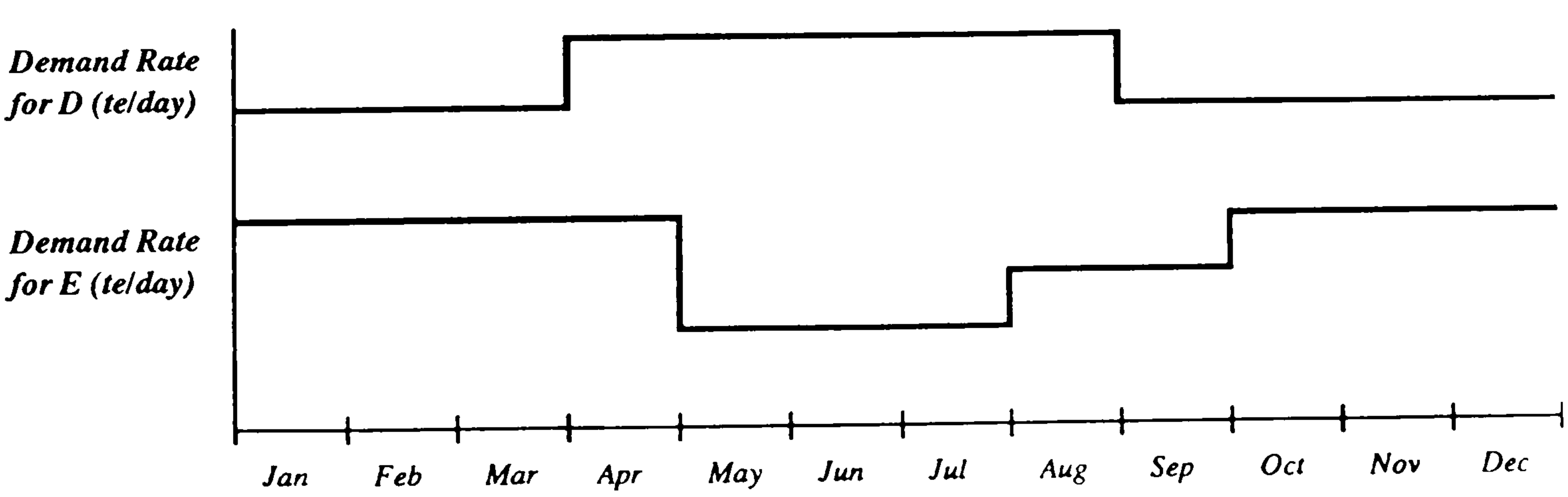
Alternatively, the product demand and the raw material supply may be expressed in terms of continuous, possibly seasonally varying, rates.

Typically, point demands may correspond to one or more specific orders, while continuous demand rates may express market predictions averaged over a large number of potential customers.

In practice, a plant may be faced with a combination of pointwise and continuous product demands and raw material supplies. This is illustrated in figure 2.6 for a plant manufacturing two end products, *D* and *E*, from three raw materials, *A*, *B* and *C*, over a horizon of one year.



(a) Pointwise Demands and Supplies



(b) Continuous Demands

Figure 2.6: Product Demands and Raw Material Supplies

It can be seen that raw materials are supplied in two large consignments at the beginning of January and June. There are three pointwise product deliveries as well as continuous demands for these products that vary seasonally.

Long campaigns often require large amounts of stocks to be held for relatively long periods of time before being used for downstream processing or sold. Inventory considerations are therefore an important aspect of plant operation.

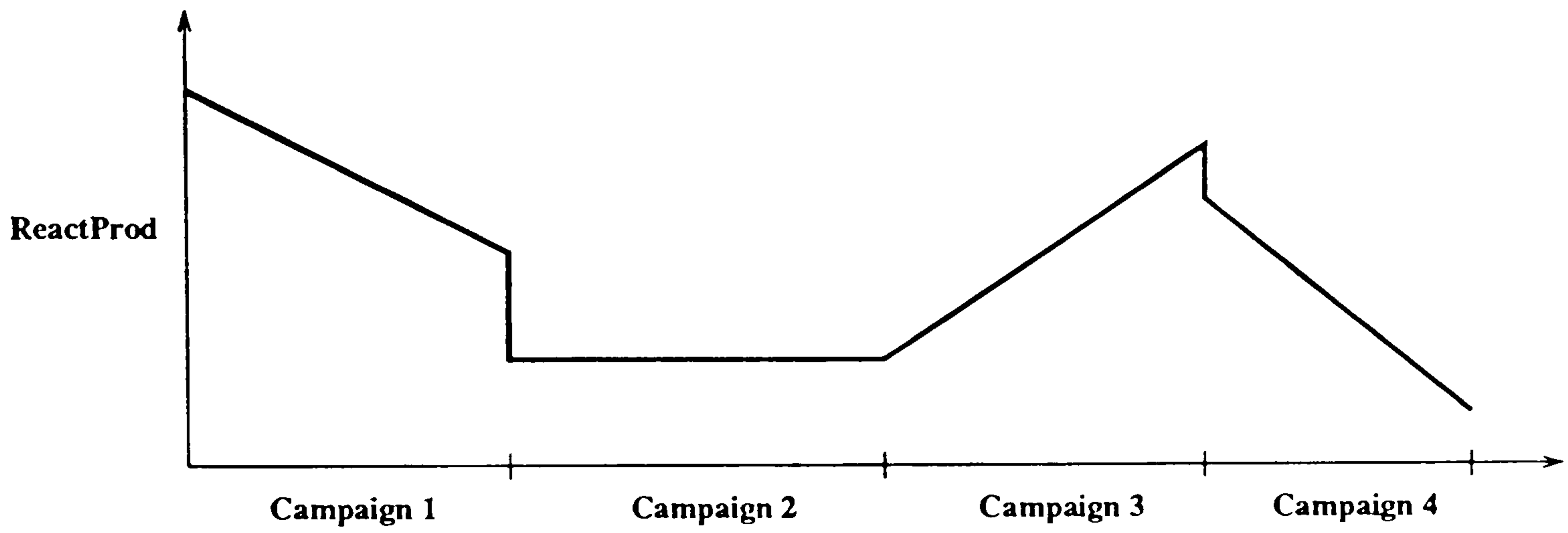


Figure 2.7: Inventory Variations

Figure 2.7 shows a typical inventory profile for a certain product over four campaigns. We can see that this is consumed during the first and fourth campaigns, while some amount is produced in the third campaign. Furthermore, discrete changes occur after the first and third campaigns due to discrete deliveries to customers.

Generally, the solution of every planning problem is characterised by trade-offs between the costs of inventories, and the costs of inter-task and inter-campaign change-overs.

2.3.3 The Campaign Planning/Scheduling Problem

The general campaign planning problem may be defined as follows:

Given:

- the time horizon of interest;
- the product recipes in terms of a STN and all its associated resource data;
- the maximum number of campaigns; and
- raw material supply and product demand data,

Determine:

- the number of active campaigns; and
- for each campaign,
    - its duration;
    - the part of the process which is active over it; and
    - a cyclic operating schedule,

so as to optimise a performance criterion (*e.g.* the maximisation of the combined production value).

2.4 Review of Literature on Campaign Planning/Scheduling

Reklaitis (1991) identifies the following three basic components of a general solution to the planning/scheduling problem of multipurpose batch plants: a production plan which involves the active campaigns and their time frames; the assignment of all the resources to the processing steps; and the detailed operating schedule (*i.e.* the task timings, the batchsizes and the flow of material through the plant and the resource utilisation profiles).

However, the resulting problem when considering all three aspects together is of high dimensionality and complexity. Most of the published work deals with simplified formulations, *e.g.* zero-wait operating mode, serial processing networks, fixed equipment/task allocation *etc.*, in order to render the above problem tractable. Comprehensive reviews of the literature on batch/semicontinuous plants have recently been given by Reklaitis (1991, 1992) and Rippin (1993).

Here, we only consider the literature on campaign planning of direct relevance to the problem described in section 2.3.3. In the next two sub-sections, hierarchical and simultaneous approaches are described.

2.4.1 Hierarchical Approaches

A number of algorithms for the optimal campaign planning problem have been presented. Mauderli and Rippin (1979) studied the combined production planning and scheduling problem developing a hierarchical procedure suitable for serial processing networks operated in a zero-wait mode. First, they consider each product individually, generating alternative production lines of a single product by assembling the available processing equipment in groups in order to achieve maximum path capacity. This is achieved by using equipment items in parallel, first in-phase and then out-of-phase, and hence removing capacity and timing stage bottlenecks. Two or more production lines for the same or different products can then be combined to construct alternative campaign candidates provided that they use disjoint subsets of the available equipment. A linear programming (LP) based screening procedure is used to determine a set of dominant campaigns. Finally, the production plan is generated by solving a LP or mixed integer linear programming (MILP) problem, allocating the available production time to the various dominant campaigns for a given set of production requirements.

The generation of the alternative production lines in the Mauderli and Rippin (1979) algorithm is based on an exhaustive enumerative procedure. A more efficient generation procedure is described by Wellons and Reklaitis (1989a) who formulated the optimal scheduling of a single-product production line as a mixed integer nonlinear programming (MINLP) model. They consider serial processing networks with zero-wait and no intermediate storage capacity policies between different processing steps. Strict batch identity is preserved and hence aggregation of different batches and/or splitting of a specific batch are not allowed. Equipment is pre-arranged into groups operating in parallel and strictly out-of-phase, where the items belonging to the same group must operate in-phase. Each group can be used at most once in each cycle in a fixed order, and each equipment item is pre-assigned to a given stage. The objective of the problem is the maximisation of the processing rate of the production line. The resulting MINLP model is nonconvex, and this, together with the large number of integer variables involved in the formulation, renders the model difficult to solve. Additionally, the proposed model exhibits high degeneracy as many path assignments result in equivalent schedules. The elimination of this degeneracy was considered by Wellons and Reklaitis (1989b) who identify a set of dominant unique path sequences and hence improve the solution efficiency of their original formulation.

A further improvement from the single-product production line scheduling to the single-product campaign formation problem has been presented by Wellons and Reklaitis (1991a). This augments the previous work (Wellons and Reklaitis, 1989a, b) to include the automatic assignment of different equipment items to groups, and also the assignment of these groups to production stages. They proposed a MINLP model along with a decomposition scheme which reduces both the integrality gap and the computational cost, and eliminates degeneracy. Their decomposition approach comprises:

- an equipment assignment master problem (MINLP) determining optimal equipment groups and thus providing an upper bound on the objective function (*e.g.* maximisation of combined production value); and

- a campaign formation subproblem (MINLP) determining the equipment assignment and production line scheduling that provides a lower bound on the objective function.

The above work was extended by Wellons and Reklaitis (1991b) to the multiple-product campaign formation problem for multipurpose batch plants. They present a procedure to identify the dominant set of multiple-product campaigns by using a linear programming (LP) procedure based on a noninferior set estimation (NISE) method. Finally, a multi-period planning model is proposed, allocating the production time among the dominant campaigns while considering simultaneously profit from sales, changeover, inventory costs and campaign set-ups.

Papageorgiou and Pantelides (1992) presented a hierarchical approach attempting to exploit better the inherent flexibility of multipurpose plants by removing various restrictions regarding the intermediate storage policies between successive processing steps, the utilisation of multiple equipment items in parallel, and also the use of the same item of equipment for more than one task within the same campaign. A three-step procedure is proposed. First, a feasible solution to the campaign planning problem subject to restrictive assumptions is obtained to determine the number of campaigns and the active parts of the original processing network involved within each campaign (*e.g.* Shah and Pantelides, 1991). Secondly, the production rate in each campaign is improved by removing these assumptions, and applying the cyclic scheduling algorithm of Shah *et al.* (1993a). Finally, the timing of the campaigns is revised to take advantage of the improved production rates (*e.g* Mauderli and Rippin, 1979). An interesting feature of this approach is that *any* existing campaign planning algorithm can be used for its first step (*e.g.* Mauderli and Rippin, 1979; Wellons and Reklaitis, 1991b; Shah and Pantelides, 1991). In contrast to previous hierarchical approaches, at the end of each step a complete feasible solution to the entire problem, if feasible, is obtained. In addition, a monotonic increase of the objective function (*e.g.* maximisation of combined production value) is achieved at the end of each step.

Although the work described above constitutes significant progress for the campaign planning problem, a number of restrictive assumptions are adopted by all of these formulations. More specifically, the following assumptions are usually made:

- Serial processing networks where every processing step has at most one sucessor and/or predecessor steps. Multiple production routes, or material recycles are not taken into account;
- Limited flexibility in the utilisation of processing equipment. Each item of processing equipment can be assigned to at most one processing step during any single campaign.

Also, all equipment items assigned to the same processing step operate in a fixed in-phase or out-of-phase manner throughout the campaign; and

- Limited operating mode (*e.g.* ZW, or NIS storage policies, no batch splitting/mixing).

2.4.2 Simultaneous Approaches

The algorithms described above are hierarchical in nature, and therefore relatively easy to implement given the reduction in the size of the problem solved at every step. On the other hand, it is difficult to relate the exact objective for each individual step in the hierarchy to the overall campaign and planning objective function, and therefore it is very difficult to assess the quality of the final solution obtained.

Shah and Pantelides (1991) proposed a single-level mathematical (MILP) formulation for the simultaneous campaign formation and planning problem. Their algorithm determines simultaneously the number and the length of the campaigns and the products and/or stable intermediates manufactured within each campaign. They consider serial processing networks operating in a mixed ZW/UIS mode, and non-identical parallel equipment items operating in-phase. The possibility of allowing the storage of stable intermediates decouples the manufacture of the relevant products into different campaigns, and therefore provides more flexibility for better equipment utilisation. Limited availability levels of utilities and storage capacity for stable intermediates being stored between campaigns may also be taken into account.

An approach suitable for general processing networks with general equipment utilisations that takes into account the possibly limited availability of resources (*e.g.* utilities, manpower, storage capacity) along with other technical and contractual constraints is described by Shah *et al.* (1993a). However, their approach is suitable only for determining cyclic operating schedules within a *single* campaign, within which all the products are manufactured.

Voudouris and Grossmann (1993) extended the work originally presented by Birewar and Grossmann (1989a, b, 1990) to campaign planning problems for multiproduct plants. They introduced cyclic scheduling, location and sizing of intermediate storage, and inventory considerations along with linearisation schemes transforming the resulting mixed integer nonlinear programming (MINLP) formulations into mixed integer linear programming (MILP) models.

Tsirukis *et al.* (1993) considered the optimal operation of multipurpose plants operating in campaign mode to fulfil outstanding orders. Resource constraints are explicitly taken into account while the limited availability of resource levels affects the operation of the plant. The problem is formulated as a mixed integer nonlinear (MINLP) model. To deal with the complexity, non-convexity and nonlinearity of the formulation, more efficient formulations along with a problem specific two-level decomposition strategy were proposed. First, a campaign formation subproblem (CFS) is solved where all customer orders are assigned to campaigns. Given the campaign structure, at the second level, equipment and resource assignment subproblem (ERAS), all the equipment items and resources are allocated to production tasks. Both subproblems are formulated as MINLP models.

Finally, Tsirukis and Reklaitis (1993a, b) proposed the use of feature extraction strategies for the efficient solution of large scale MINLP problems in order to direct the optimisation methods towards the most promising regions of the original problem domain.

2.5 Conclusions

As is evident from the preceding literature review, a substantial amount of work has been done on the solution of the campaign planning/scheduling problem. However many features of the problem, such as the handling of complex processing recipes and the exploitation of flexible resources, have not yet been addressed to a satisfactory extent.

A general framework for the description of batch processes and the associated scheduling problems has been presented by Kondili *et al.* (1988, 1993). The framework, based on the State-Task Network (STN) concept, has already been applied successfully to the mathematical formulation and solution of short-term (Kondili *et al.*, 1993; Shah *et al.*, 1993b) and cyclic (Shah *et al.*, 1993a) scheduling problems.

One of the major aims of this work is to extend the STN framework to cover general campaign planning and scheduling problems. In the next chapter, we present the mathematical formulation of the problem, while issues pertaining to the solution of the resulting large mixed integer optimisation problem are considered in chapters 4 and 5.

Chapter 3

Mathematical Formulation of the Campaign Planning/Scheduling Problem

This chapter presents a mathematical formulation for the general campaign planning/scheduling problem defined in the previous chapter. In contrast to hierarchical approaches, a *single-level* formulation is developed encompassing both overall planning considerations pertaining to the campaign structure and scheduling constraints describing the detailed operation of the plant during each campaign. However, in the interests of clarity, the two types of constraint are presented separately in sections 3.2 and 3.3 respectively. Finally, the objective function for this complex optimisation problem is discussed in section 3.4.

3.1 Basis of the Mathematical Formulation

This section describes the basic characteristics and major assumptions underlying our mathematical formulation.

- All product recipes are given in terms of STNs. Resource characteristics and functionalities, and production requirements are also assumed to be known.
- We assume a maximum potential number of campaigns, N, not all of which need be active in the final solution. The number of active campaigns and the processing tasks which are involved within each campaign are to be determined by the optimisation algorithm.
- We assume cyclic plant operation with a given cycle time within each campaign. The formulation aims to determine a detailed cyclic operating schedule based on the discrete time representation of Kondili *et al.* (1993) and Shah *et al.* (1993a, b).
- We ignore start-up and shut-down periods in each campaign assuming that they are negligible compared to the campaign length. The latter, comprising an integer number of cycles, is to be determined.
- Product demands may be incurred either at the end of, or during the planning horizon. Demands during the horizon may either fall due at distinct points or be expressed as continuous rates of demand. Such rates may be different for different periods of the horizon. A similar set of assumptions is adopted for raw material supplies during the horizon of interest.

- Changes in resource availability and/or prices may occur at a set of discrete points over the horizon.
- All data are assumed to be deterministic.

3.2 Mathematical Formulation — Campaign Planning Constraints

We present the mathematical formulation in two parts. The first refers to campaign planning constraints while the second deals with the constraints describing a cycle in the cyclic operating schedule of each campaign.

As has already been mentioned, a number of discrete events (*e.g.* product deliveries, changes in utility availability and/or price) may take place during the horizon. We sort the prespecified *NDP distinct* times (τ_i) at which these events occur in ascending order such that $\tau_1 < \tau_2 \cdots < \tau_{NDP}$ and require each one of them to coincide with an inter-campaign boundary as shown in figure 3.1. As nothing prevents two successive campaigns from being identical, this assumption is not restrictive provided a sufficiently large number of campaigns is considered .

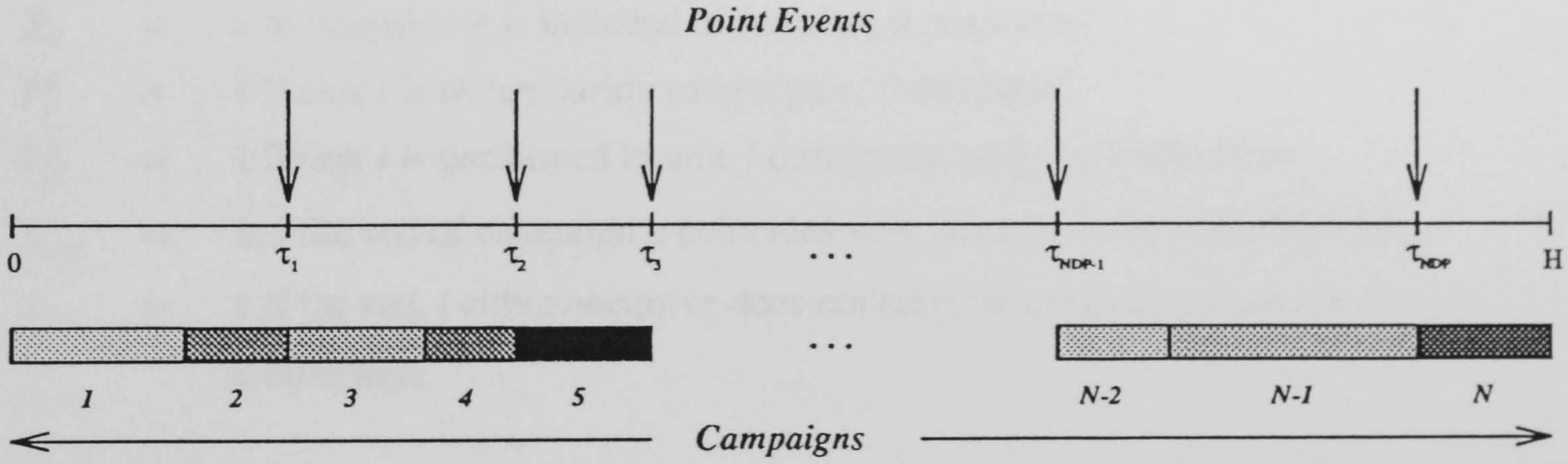


Figure 3.1: Point Events and Campaigns

A number of parameters and sets are associated with the campaign planning problem. These are listed below; their precise meaning will become clear later in this section.

The time horizon of interest is denoted by H.

Campaign c is characterised by:

$D^{\min}$, $D^{\max}$	:	minimum and maximum acceptable campaign durations.
T_c	:	cycle time of campaign c.
Ψ	:	maximum acceptable campaign complexity.

Task i is characterised by:

K_i : set of equipment items that are suitable for task i.

ψ_i : complexity factor associated with task i.

State s is characterised by:

C_s : maximum storage capacity for state s.

Unit j is characterised by:

I_j : set of tasks for which unit j is suitable.

n_j : the maximum number of tasks for which unit j can be used in any single campaign.

Utility u is characterised by:

$U_{uc}^{\max}$: maximum availability of utility u over campaign c.

We postulate a maximum number, N, of campaigns to be considered. The formulation is based on the following key variables characterising each campaign $c = 1..N$:

Binary Variables:

X_c = 1 if campaign c is included in the plan; 0 otherwise.

Y_i^c = 1 if task i is active during campaign c; 0 otherwise.

Y_{ij}^c = 1 if task i is performed in unit j during campaign c; 0 otherwise.

$\Lambda_{\mu c}$ = 1 if the end of campaign c coincides with distinct point μ; 0 otherwise.

λ_{ic} = 1 if the task i either occurs or does not occur in *both* campaigns c and $c+1$; 0 otherwise.

Integer Variables:

n_c = number of cycles in campaign c.

Continuous Variables:

ΔS_{sc} = nett accumulation of state s within each cycle in campaign c.

BS_{sc}^I = amount of material stored in state s at the start of campaign c.

BS_{sc}^F = amount of material stored in state s at the end of campaign c *before* any discrete deliveries and receipts.

$BS_{sc}^{F'}$ = amount of material stored in state s at the end of campaign c *after* any discrete deliveries and receipts.

$R_{s\mu}$ = amount of material in state s to be received from external suppliers at time τ_μ.

$D_{s\mu}$ = amount of material in state s to be delivered to external customers at time τ_μ.

r_{sc} = rate of continuous receipt of material s during campaign c.

d_{sc} = rate of continuous delivery of material s during campaign c.

t_c = starting time of campaign c.

Θ_c = duration of inter-campaign change-over between campaigns c and $c+1$.

With the exception of ΔS_{sc}, all variables are assumed to be non-negative.

3.2.1 Campaign Timing Constraints

The campaign duration is equal to the number of cycles within each campaign n_c multiplied by the cycle time T_c. This duration must lie within specified bounds if the campaign is included in the final plan (*i.e.* $X_c = 1$); otherwise it must be zero.

$$D^{\min} X_c \le n_c T_c \le D^{\max} X_c \quad \forall\, c \tag{3.1}$$

The starting times of the active campaigns are governed by the following constraints:

$$t_{c+1} \ge t_c + n_c T_c + \Theta_c \quad \forall\, c = 1 .. N-1 \tag{3.2a}$$

and

$$t_N + n_N T_N \le H \tag{3.2b}$$

where Θ_c is the duration of the inter-campaign change-over needed for reconfiguring the plant between campaigns c and $c+1$. This may be constant, or it may depend on the degree of similarity between two consecutive campaigns (see section 3.2.8). Note that t_1 coincides with the start of the horizon, being equal to zero.

The general integer variable n_c may be converted to a set of binary variables. If $n_c^{\max}$ is an upper bound on n_c, then the latter can be substituted by the expression:

$$n_c = \sum_{i=0}^{r_c - 1} 2^i L_{ic} \quad \forall\, c \tag{3.3}$$

where the L_{ic} are 0-1 variables and $r_c = \lfloor \log_2 n_c^{\max} \rfloor + 1$, *i.e.* $2^{r_c - 1} \le n_c^{\max} < 2^{r_c}$. Thus the L_{ic} variables provide a binary encoding of the general integer n_c.

3.2.2 Campaign Activity Constraints

The actual number of campaigns in the optimal schedule will often be smaller than the maximum allowed. In such cases, there will be significant degeneracy as all solutions with the same number of campaigns are equivalent. The degeneracy can be removed by forcing all inactive campaigns to the end of the horizon:

$$X_{c+1} \leq X_c \quad \forall\, c = 1..N-1 \tag{3.4}$$

These constraints ensure that a campaign can be active only if all the previous ones are also active.

No task i should be included in campaign c if that campaign is not active, *i.e.* $X_c = 0$. Therefore, we obtain:

$$Y_i^c \leq X_c \quad \forall\, c, i \tag{3.5a}$$

Also, if a task is not included in a campaign, then no equipment item need be used for it:

$$Y_{ij}^c \leq Y_i^c \quad \forall\, c, i, j \in K_i \tag{3.5b}$$

3.2.3 Operability and Safety Constraints

Various constraints reflecting plant operability and safety can be imposed on the desired campaign plan. For instance, if a set Ξ of tasks are mutually exclusive over the same campaign, we can simply write:

$$\sum_{i \in \Xi} Y_i^c \leq X_c \quad \forall\, c \tag{3.6}$$

This ensures that, if campaign c is active, then at most one task in Ξ can take place in it.

Operational complexity may be limited by characterising each task i by a measure of complexity ψ_i, and then imposing an upper limit Ψ on the complexity of any campaign c:

$$\sum_i \psi_i\, Y_i^c \leq \Psi X_c \quad \forall\, c \tag{3.7}$$

The above operability and safety constraints are concerned with whether or not a certain task i is active over a certain campaign c. In some cases, we may have further constraints on the utilisation of specific equipment items j. For instance, the imposition of an upper bound n_j on the number of different tasks for which an item j can be used during any single campaign can simply be written as:

$$\sum_{i \in I_j} Y_{ij}^c \leq n_j X_c \quad \forall\ c,\ j \tag{3.8}$$

In particular, by specifying $n_j = 1$, we can revert to the common "unique equipment assignment" case (Reklaitis, 1991).

3.2.4 Interior Point Event Constraints

It is often assumed in formulating planning and scheduling problems that all feedstocks are available at the beginning of the plant operation, and all products are delivered at the end of the planning horizon. However, in realistic market conditions, raw materials may be received at specific dates and products may be delivered before the end of the planning horizon continuously or at prespecified distinct time points (*e.g.* seasonal products).

Also, resource availabilities may vary during the planning horizon due to many reasons, such as the contractual nature of utility provision, machine reconfiguration, cleaning *etc.*

Suppose that all pointwise changes occur at a finite set of distinct time points τ_μ, $\mu \in 1..NDP$. Our formulation is based on the idea of forcing all distinct points to coincide with the end of a campaign. The following constraints relate the distinct points τ_μ to the campaign timing variables t_c:

$$\sum_c \Lambda_{\mu c} = 1 \quad \forall\ \mu \tag{3.9a}$$

$$\sum_\mu \Lambda_{\mu c} \leq X_c \quad \forall\ c \tag{3.9b}$$

and

$$H(1 - \sum_\mu \Lambda_{\mu c}) \geq t_c + n_c T_c - \sum_\mu \tau_\mu \Lambda_{\mu c} \geq 0 \quad \forall\ c \tag{3.9c}$$

Note that constraints (3.9a) demand that every distinct point coincide with the end of exactly one campaign, whilst constraints (3.9b) allow the end of each active campaign ($X_c = 1$) to coincide with at most one distinct point. Finally, constraints (3.9c) relate the distinct points τ_μ to the campaign timings. It should be noted that, if no distinct point coincides with the end of campaign c, *i.e.* $\Lambda_{\mu c} = 0,\ \forall\ \mu$, then constraints (3.9c) are inactive and redundant with respect to constraints (3.2a) and (3.2b). On the other hand, if one of the $\Lambda_{\mu c}$ is 1 for a given campaign c, then (3.9c) are equivalent to:

$$\tau_\mu = t_c + n_c T_c$$

thereby aligning the end of campaign c with discrete event point μ.

3.2.5 Material Balance Constraints over a Campaign

Every campaign c is characterised by three continuous variables, BS^I_{sc}, BS^F_{sc} and $BS^{F'}_{sc}$ corresponding to inventory of material s at the start, and at the end *before* and *after* discrete deliveries/receipts, respectively, of the campaign, as illustrated in figure 3.2.

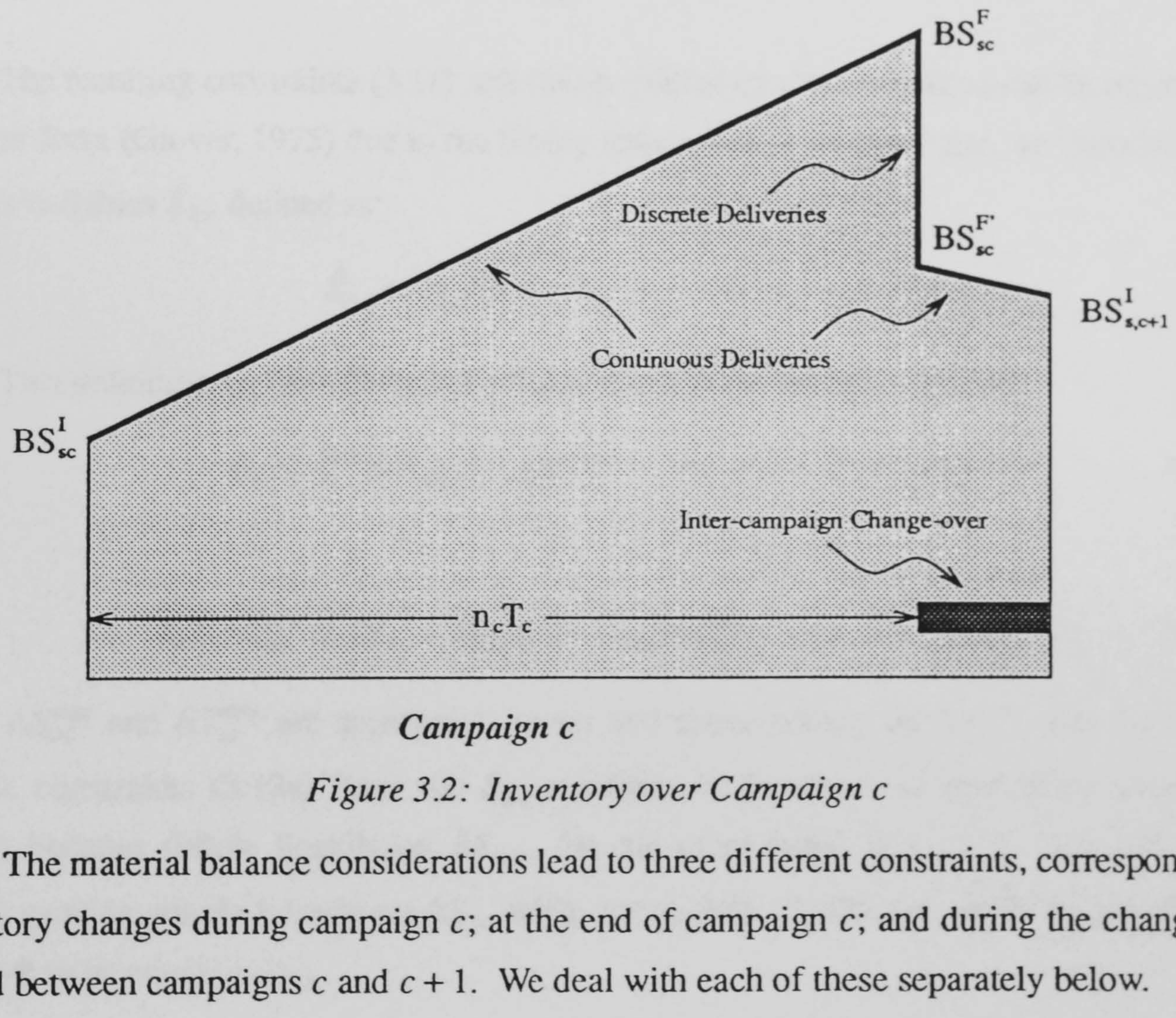


Figure 3.2: Inventory over Campaign c

The material balance considerations lead to three different constraints, corresponding to inventory changes during campaign c; at the end of campaign c; and during the change-over period between campaigns c and $c+1$. We deal with each of these separately below.

3.2.5.1 Material Balance during Campaign c

Ignoring for the moment product deliveries and raw material receipts, the amount of material BS^F_{sc} in state s at the end of campaign c is equal to the amount at the start of the campaign BS^I_{sc} plus the nett accumulation, $n_c \Delta S_{sc}$, during the campaign:

$$BS^F_{sc} = BS^I_{sc} + n_c \, \Delta S_{sc} \quad \forall \, c, s \tag{3.10}$$

The initial amount of material, BS^I_{s1}, for each state at the start of the time horizon is assumed to be given.

It may be worth mentioning that in their periodic scheduling formulation, Shah *et al.* (1993a) assume no accumulation of intermediates within a cycle. The main advantage of constraints (3.10) over this simplification is that they take explicit account of the true inventory of every state at the end of each campaign without the need for any distinction between different types of material.

We note that constraints (3.10) involve a nonlinear term in the right hand side of the equation. In order to maintain linearity in our formulation, we must replace this term with a linear one. From constraints (3.10) and (3.3), we obtain:

$$BS_{sc}^{F} = BS_{sc}^{I} + \sum_{i=0}^{r_c-1} 2^i \ L_{ic} \Delta S_{sc} \quad \forall \ c, s \tag{3.11}$$

The resulting constraints (3.11) still involve bilinear terms, but these can be represented in linear form (Glover, 1975) due to the binary nature of L_{ic}. In particular, we introduce continuous variables $\hat{S}_{ics}$ defined as:

$$\hat{S}_{ics} \equiv L_{ic} \Delta S_{sc} \quad \forall \ c, s, i = 0..r_c - 1$$

This definition can be effected implicitly through the linear constraints:

$$L_{ic} \Delta S_{sc}^{\min} \leq \hat{S}_{ics} \leq L_{ic} \Delta S_{sc}^{\max} \quad \forall \ c, s, i = 0..r_c - 1 \tag{3.12a}$$

and

$$(1 - L_{ic}) \Delta S_{sc}^{\min} \leq \Delta S_{sc} - \hat{S}_{ics} \leq (1 - L_{ic}) \Delta S_{sc}^{\max} \quad \forall \ c, s, i = 0..r_c - 1 \tag{3.12b}$$

where $\Delta S_{sc}^{\min}$ and $\Delta S_{sc}^{\max}$ are appropriate lower and upper bounds on ΔS_{sc}[1]. For the case of $L_{ic} = 0$, constraints (3.12a) force the $\hat{S}_{ics}$ variables to the value of zero while constraints (3.12b) become simple bounds on ΔS_{sc}. On the other hand, if $L_{ic} = 1$, then constraints (3.12a) provide simple bounds on ΔS_{sc} while constraints (3.12b) are active forcing the $\hat{S}_{ics}$ variables to be equal to ΔS_{sc}.

If continuous receipt or delivery of material s *do* take place during campaign c, then (3.10) must be modified to:

$$BS_{sc}^{F} = BS_{sc}^{I} + n_c \ \Delta S_{sc} + n_c T_c (r_{sc} - d_{sc}) \quad \forall \ c, s \tag{3.10'}$$

where r_{sc} and d_{sc} are the corresponding receipt and delivery rates.

Once again, we replace n_c by its binary encoding (3.3), introducing the new variables:

$$\hat{S}_{ics} \equiv L_{ic} (\Delta S_{sc} + T_c (r_{sc} - d_{sc})) \quad \forall \ c, s, i = 0..r_c - 1$$

[1] Procedures for deriving such bounds, the tightness of which has a significant effect on solution efficiency, are described in Appendix A.

defined through a modification of (3.12):

$$L_{ic}\left(\Delta S_{sc}^{\min} + T_c(r_{sc}^{\min} - d_{sc}^{\max})\right) \leq \hat{S}_{ics} \leq L_{ic}\left(\Delta S_{sc}^{\max} + T_c(r_{sc}^{\max} - d_{sc}^{\min})\right)$$

$$\forall\, c, s, i = 0..r_c - 1 \qquad (3.12a')$$

and

$$(1 - L_{ic})\left(\Delta S_{sc}^{\min} + T_c(r_{sc}^{\min} - d_{sc}^{\max})\right) \leq \Delta S_{sc} + T_c(r_{sc} - d_{sc}) - \hat{S}_{ics} \leq (1 - L_{ic})\left(\Delta S_{sc}^{\max} + T_c(r_{sc}^{\max} - d_{sc}^{\min})\right)$$

$$\forall\, c, s, i = 0..r_c - 1 \qquad (3.12b')$$

where $\left[r_{sc}^{\min}, r_{sc}^{\max}\right]$, and $\left[d_{sc}^{\min}, d_{sc}^{\max}\right]$ are appropriate ranges for r_{sc} and d_{sc} respectively.

In any case, using the new variables $\hat{S}_{ics}$ (defined through (3.12) or (3.12′)), we can rewrite (3.11) in linear form:

$$BS_{sc}^{F} = BS_{sc}^{I} + \sum_{i=o}^{r_c-1} 2^i\, \hat{S}_{ics} \quad \forall\, c, s \qquad (3.13a)$$

3.2.5.2 Material Balance at End of Campaign c

As has already been mentioned, BS_{sc}^{F} is the inventory of state s at the end of campaign c *before* any discrete deliveries or receipts take place. The inventory *after* such changes, $BS_{sc}^{F'}$, is related to it through:

$$BS_{sc}^{F'} = BS_{sc}^{F} + \sum_{\mu} \Lambda_{\mu c}(R_{s\mu} - D_{s\mu}) \quad \forall\, c, s \qquad (3.13b)$$

It should be noted that $BS_{sc}^{F'}$ differs from Bs_{sc}^{F} only if a discrete event μ coincides with the end of campaign c (*i.e.* $\Lambda_{\mu c} = 1$).

The receipts and deliveries $R_{s\mu}$ and $D_{s\mu}$ may be constant or be allowed to vary within given bounds. In the latter case they must be treated as unknowns, and this introduces a bilinearity in (3.13b). However, this again involves the product of binary and continuous variables and can therefore be reformulated exactly to linear form (Glover, 1975).

3.2.5.3 Material Balance during Inter-campaign Change-over Periods

During the inter-campaign change-over periods, no production takes place but inventory changes may still occur due to continuous receipts and/or deliveries of material. Thus the inventory at the start of campaign $c + 1$ is related to that at the end of campaign c through:

$$BS^{I}_{s,c+1} = BS^{F'}_{sc} + (r_{s,c+1} - d_{s,c+1})\,\Theta_c \quad \forall\, c,\, s \tag{3.13c}$$

We note that, in common to all other discrete events, any discrete changes in the continuous receipt and delivery rates are assumed to coincide with campaign *ends*. Therefore, the rates obtaining during the inter-campaign change-over period are those corresponding to the *next* campaign $c+1$, namely $r_{s,c+1}$ and $d_{s,c+1}$.

We also note that, in principle, (3.13c) contains a bilinear term involving two continuous quantities. Approximate linearisation techniques (*e.g.* McCormick, 1976) may be applied to deal with this complication. However, the above linearisations are not necessary if *either* the receipt/delivery rates *or* the change-over duration are known constants.

3.2.6 Storage Capacity and Inventory Constraints

Here we consider various constraints imposed by the limited availability of storage capacity.

For states s stored in *dedicated* storage facilities, we have:

$$0 \le BS^{I}_{sc} \le C_s \quad \forall\, c,\, s \tag{3.14a}$$

$$0 \le BS^{F}_{sc} \le C_s \quad \forall\, c,\, s \tag{3.14b}$$

and

$$0 \le BS^{F'}_{sc} \le C_s \quad \forall\, c,\, s \tag{3.14c}$$

As mentioned earlier in chapter 2, an alternative storage mode is that of *shared* intermediate storage where various states can share the same storage facility at the same time (*e.g.* warehouses). Consider a storage resource w that can store a set of states S_w with a maximum available capacity of C_w. The cumulative storage level of the suitable states must not exceed the available storage capacity of the storage facility:

$$\sum_{s \in S_w} \omega_{ws} BS^{I}_{sc} \le C_w \quad \forall\, c,\, w \tag{3.15a}$$

$$\sum_{s \in S_w} \omega_{ws} BS^{F}_{sc} \le C_w \quad \forall\, c,\, w \tag{3.15b}$$

and

$$\sum_{s \in S_w} \omega_{ws} BS^{F'}_{sc} \le C_w \quad \forall\, c,\, w \tag{3.15c}$$

where ω_{ws} is the amount of storage resource w required for storing a unit amount of s.

Irrespective of the precise mode of storage, it may be necessary to maintain certain minimum amounts of inventory (*e.g.* to provide a safety margin against uncertainty in demands) for certain products. Furthermore, safety and other considerations may lead to the imposition of upper bounds on the inventory of certain materials. This leads to bounds of the form:

$$BS_s^{\min} \leq BS_{sc}^{I} \leq BS_s^{\max} \quad \forall\ c,\ s \tag{3.16a}$$

$$BS_s^{\min} \leq BS_{sc}^{F} \leq BS_s^{\max} \quad \forall\ c,\ s \tag{3.16b}$$

and

$$BS_s^{\min} \leq BS_{sc}^{F'} \leq BS_s^{\max} \quad \forall\ c,\ s \tag{3.16c}$$

As will be seen later (section 3.4.5), inventory variation is approximately linear, and therefore these constraints also guarantee feasibility at all times *during* the campaign and inter-campaign change-over periods.

3.2.7 Seasonal Variations

Several types of data defining the campaign planning problem may be seasonally dependent. These include the utility availability and the rates of continuous product demand and raw material supply.

3.2.7.1 Utility Availability Variations

If the ceiling of utility availability, $U_{uc}^{\max}$, changes during the planning horizon, then additional constraints need to be written to describe this profile. In particular, we assume that the maximum availability of utility u is a piecewise constant function over the time partition defined by $\{\tau_\mu\}$. Let the initial availability of utility u be $U_{u0}^{\max}$, and let $\Delta U_{u\mu}$ denote the *change* in this availability at time τ_μ as illustrated in figure 3.3.

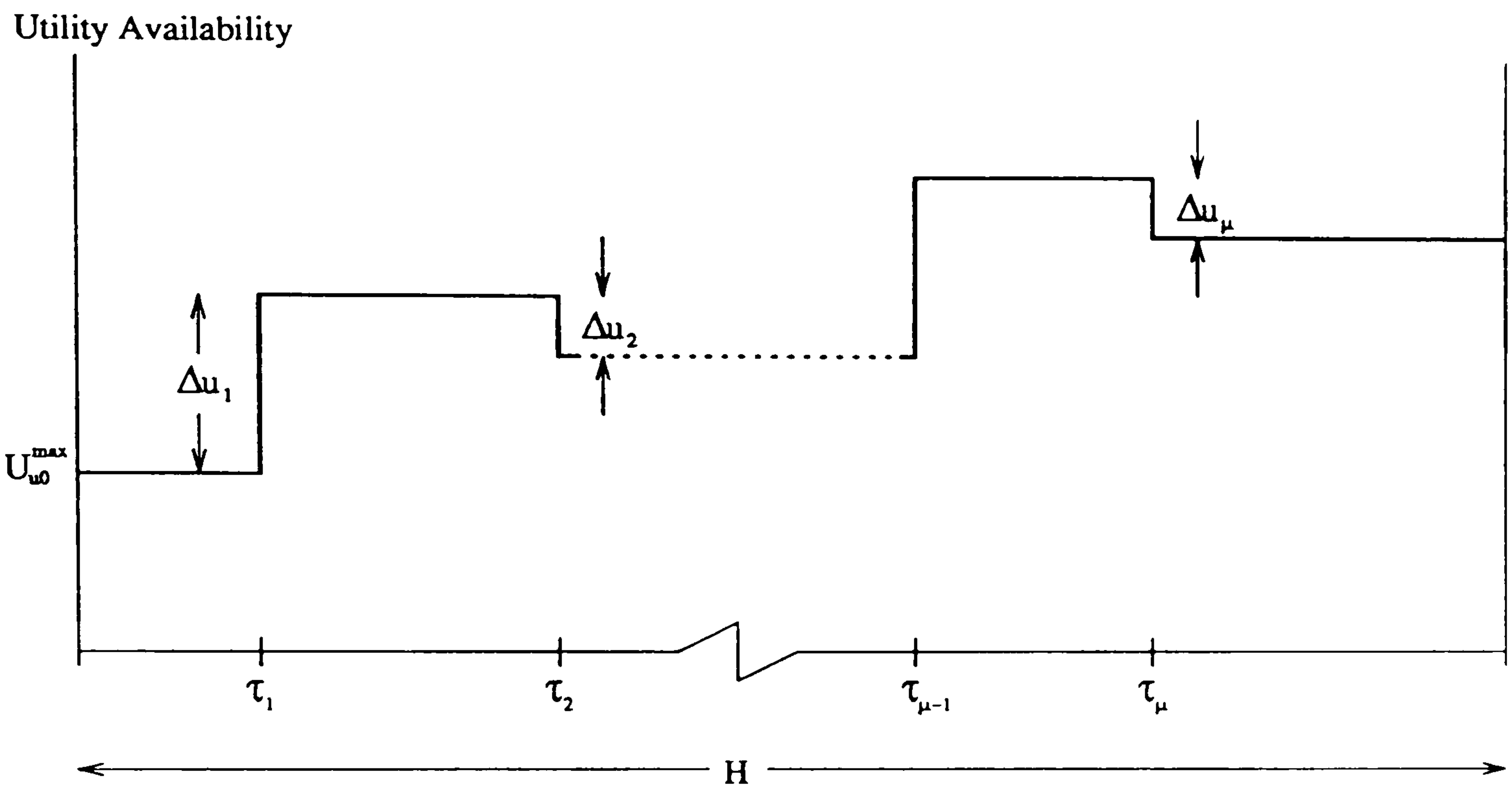


Figure 3.3: Variation of Utility Availability Level over Time Horizon

Then the maximum availability of utility u over campaign c, U_{uc}^{max}, is a variable given by:

$$U_{u,c+1}^{max} = U_{uc}^{max} + \sum_{\mu} \Lambda_{\mu c} \, \Delta U_{u\mu} \quad \forall\, c, u \tag{3.17}$$

If the end of campaign c coincides with the μth discrete event, then these constraints ensure that $U_{u,c+1}^{max} = U_{uc}^{max} + \Delta U_{u\mu}$. Otherwise, $U_{u,c+1}^{max} = U_{uc}^{max}$, *i.e.* the utility availability remains the same.

3.2.7.2 Product Demand and Raw Material Supply Variations

As already discussed in section 3.2.5.1, the rates of continuous product demand and/or raw material supply, may be given constants or may vary between given lower and upper bounds. The latter, moreover, may seasonally dependent — a very common situation in certain sectors such as the food industry.

We assume that the demand and supply bounds vary in a piecewise constant manner over the time horizon, and, as all discrete events in our formulation, any changes in them are aligned with the campaign ends. In a manner similar to section 3.2.7.1, the following constraints are formulated:

$$r_{s,c+1}^{max} = r_{sc}^{max} + \sum_{\mu} \Lambda_{\mu c} \Delta r_{s\mu}^{max} \quad \forall\, c, s \tag{3.18a}$$

$$r_{s,c+1}^{min} = r_{sc}^{min} + \sum_{\mu} \Lambda_{\mu c} \Delta r_{s\mu}^{min} \quad \forall\, c, s \tag{3.18b}$$

$$d_{s,c+1}^{\max} = d_{sc}^{\max} + \sum_{\mu} \Lambda_{\mu c} \Delta d_{s\mu}^{\max} \quad \forall\ c,\ s \tag{3.18c}$$

and

$$d_{s,c+1}^{\min} = d_{sc}^{\min} + \sum_{\mu} \Lambda_{\mu c} \Delta d_{s\mu}^{\min} \quad \forall\ c,\ s \tag{3.18d}$$

where $\Delta x_{s\mu}$ generally denotes the change in quantity x caused by point event μ. The bounds are used to constrain the actual receipt and delivery rates in each campaign:

$$r_{sc}^{\min} \leq r_{sc} \leq r_{sc}^{\max} \quad \forall\ c,\ s \tag{3.19a}$$

and

$$d_{sc}^{\min} \leq d_{sc} \leq d_{sc}^{\max} \quad \forall\ c,\ s \tag{3.19b}$$

Of course, in many applications the actual receipt or delivery rates may be known a priori but still be seasonally dependent. In such cases, the upper and lower bounds are identical, and (3.18) and (3.19) collapse to:

$$r_{s,c+1} = r_{sc} + \sum_{\mu} \Lambda_{\mu c} \Delta r_{s\mu} \quad \forall\ c,\ s \tag{3.20a}$$

and

$$d_{s,c+1} = d_{sc} + \sum_{\mu} \Lambda_{\mu c} \Delta d_{s\mu} \quad \forall\ c,\ s \tag{3.20b}$$

3.2.8 Inter-campaign Change-over Constraints

The duration of the inter-campaign change-over period may depend on the nature of the two consecutive campaigns, and more specifically on how different they are. The need for a suitable measure denoting the degree of similarity of two successive campaigns is evident.

In order to formulate mathematically the difference between consecutive campaigns, we introduce a *Similarity Measure* (SM_c) between campaigns c and $c+1$. This is defined as follows:

$$SM_c = \frac{\sum_i \hat{\lambda}_{ic}}{NT} \quad \forall\ c = 1..N-1 \tag{3.21}$$

where NT is the total number of processing tasks in the problem. It will be recalled that $\hat{\lambda}_{ic}$ is one if task i is involved in both or neither of campaigns c and $c+1$. Thus a SM of 1 denotes two campaigns which involve two completely identical sets of tasks, whereas a *SM* of 0 corresponds to two entirely different campaigns. As $SM_c \rightarrow 1$, the two consecutive campaigns, c and $c+1$, are increasingly similar which will normally imply a smaller duration of inter-campaign change-over.

If both c and $c+1$ are active, then we assume that the duration of the change-over will, in general, comprise a fixed term Θ^F and a variable term, Θ^V, proportional to $1 - SM_c$, *i.e.* their degree of *dis*similarity:

$$\Theta_c = \Theta^F + (1 - SM_c)\,\Theta^V \quad \forall\, c = 1..N-1$$

where Θ^F and Θ^V are given constants. However, if campaign $c+1$ is inactive, it is reasonable to assume that no change-over is incurred. We therefore modify the above to:

$$X_{c+1}\,(\Theta^F + \Theta^V) \geq \Theta_c \geq X_{c+1}\,\Theta^F + (X_{c+1} - SM_c)\,\Theta^V \quad \forall\, c = 1..N-1 \qquad (3.22)$$

We note that if $X_{c+1} = 0$ (*i.e.* campaign $c+1$ is not active), (3.22) will force the non-negative variable Θ_c to zero. On the other hand, if $X_{c+1} = 1$, then it becomes:

$$(\Theta^F + \Theta^V) \geq \Theta_c \geq \Theta^F + (1 - SM_c)\,\Theta^V \quad \forall\, c = 1..N-1$$

and the optimisation will normally force the second part of this inequality to be active at the solution.

We now have to define the variables λ_{ic} appearing in (3.21). This is done through the constraints:

$$\lambda_{ic} \geq 1 - Y_i^c - Y_i^{c+1} \quad \forall\, i,\, c = 1..N-1 \qquad (3.23a)$$

$$\lambda_{ic} \geq Y_i^c + Y_i^{c+1} - 1 \quad \forall\, i,\, c = 1..N-1 \qquad (3.23b)$$

and

$$\lambda_{ic} \leq Y_i^c - Y_i^{c+1} + 1 \quad \forall\, i,\, c = 1..N-1 \qquad (3.24a)$$

and

$$\lambda_{ic} \leq Y_i^{c+1} - Y_i^c + 1 \quad \forall\, i,\, c = 1..N-1 \qquad (3.24b)$$

Note that constraints (3.23) force the λ_{ic} variables to one only if $Y_i^c = Y_i^{c+1}$, while (3.24) ensure that $\lambda_{ic} = 0$ if $Y_i^c \neq Y_i^{c+1}$. Furthermore, the λ_{ic} can be treated as continuous variables as their integrality is implied by constraints (3.23) and (3.24), and the integrality of the Y_i^c variables.

3.3 Mathematical Formulation — Campaign Scheduling Constraints

The constraints presented in the last section deal with the overall campaign structure, namely the number and duration of the campaigns and the tasks which are active in each campaign. They also characterise the inventories at the campaign boundaries, and the utility availabilities and demand and supply rates over each campaign.

We now turn our attention to the detailed scheduling of the plant operation in each campaign. As has already been mentioned, a campaign is made up of a number of cycles as shown in figure 3.4. Each cycle involves the same pattern of operations which are repeated throughout the campaign at a fixed cycle time T_c. For the purposes of the formulation we ignore the end-effects of starting up and shutting down each campaign, effectively assuming that $n_c \gg 1$.

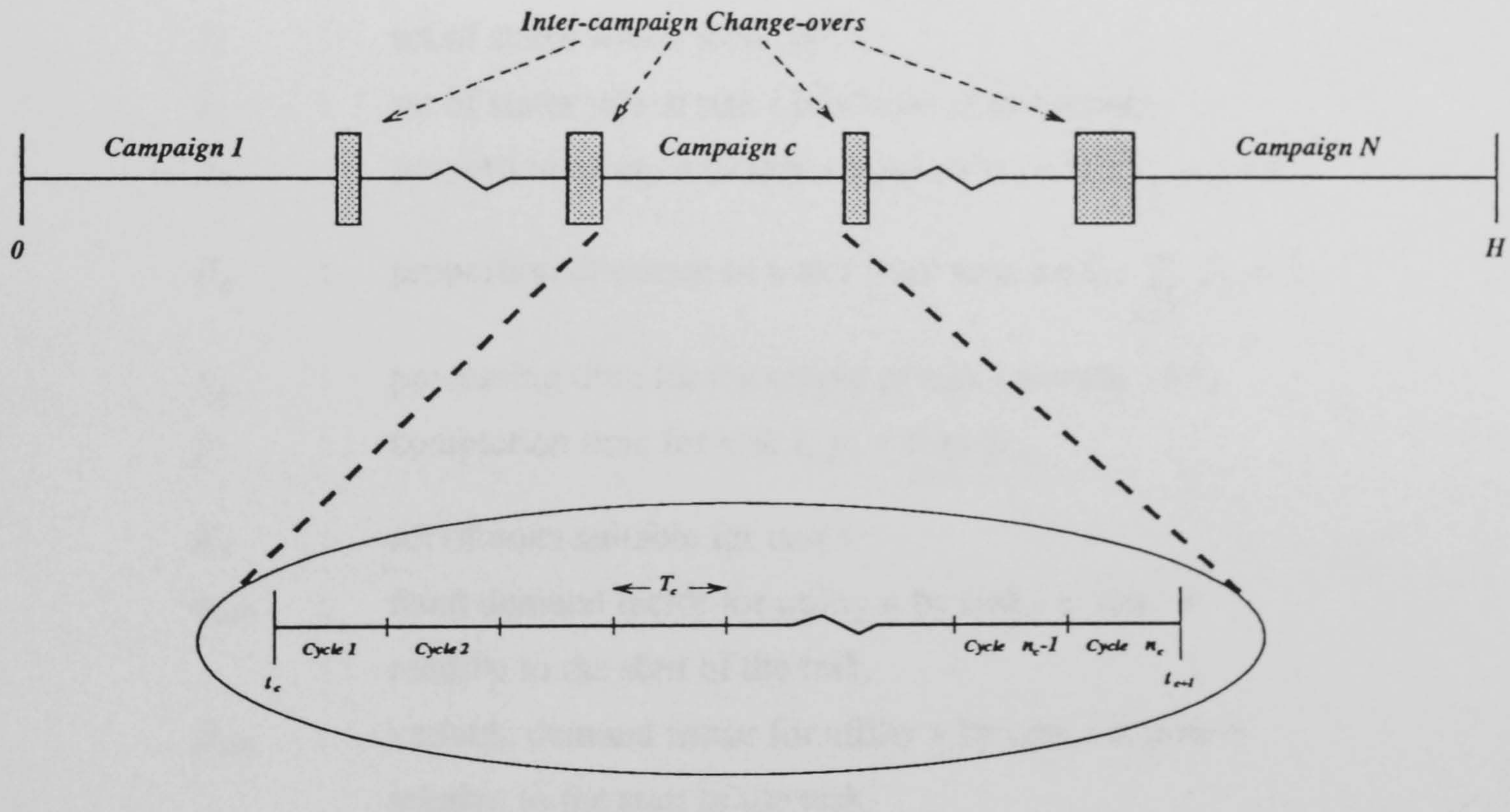


Figure 3.4: Cyclic Nature of Campaigns

As argued by Shah *et al.* (1993a), in the case of general multipurpose plants the cycle time is determined by the level of operating complexity that the plant operator is willing to accept rather than being the result of any technical considerations. Thus, a cycle time of 1 week generally entails more complexity than one of 24 hours as there is a longer sequence of operations to be executed before a regular pattern is established. On the other hand, a longer cycle time usually implies better plant performance although this is not always true (for instance, in processes involving many zero-wait operations, a longer cycle time may result in more idle time per cycle). In any case, we assume that the cycle time T_c is given but the precise operating schedule for each campaign is to be determined.

In common to other STN-based formulations, we adopt a discrete representation of time dividing the cycle time T_c into a number of time intervals of equal duration as shown in figure 3.5:

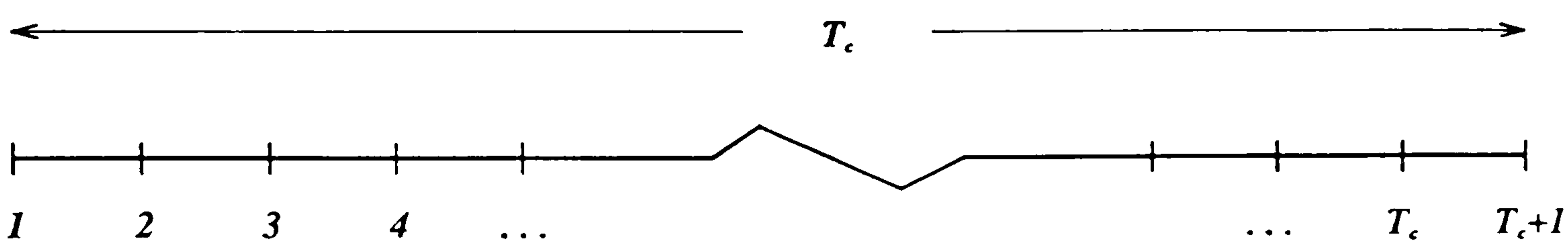


Figure 3.5: Discrete Representation of Time

The number of such intervals will henceforth also be denoted by T_c.

A number of parameters and sets are associated with the campaign scheduling problem. These listed below; their precise meaning will become clear later in this section.

Task i is characterised by:

S_i : set of states which feed task i.

$\bar{S}_i$: set of states which task i produces as its outputs.

ρ_{is} : proportion of input of task i from state $s \in S_i$; $\sum_{s \in S_i} \rho_{is} = 1$.

$\bar{\rho}_{is}$: proportion of output of task i from state $s \in \bar{S}_i$; $\sum_{s \in \bar{S}_i} \bar{\rho}_{is} = 1$.

p_{is} : processing time for the output of task i to state $s \in \bar{S}_i$.

p_i : completion time for task i, $p_i = \max_{s \in \bar{S}_i} p_{is}$.

K_i : set of units suitable for task i.

$\alpha_{ui\theta}$: fixed demand factor for utility u by task i at time θ relative to the start of the task.

$\beta_{ui\theta}$: variable demand factor for utility u by task i at time θ relative to the start of the task.

State s is characterised by:

T_s : set of tasks receiving material from state s.

$\bar{T}_s$: set of tasks producing material in state s.

C_s : maximum storage capacity for state s.

K_s : set of units suitable for state s.

Unit j is characterised by:

I_j : set of tasks (including cleaning) for which unit j is suitable.

V_j : capacity of unit j.

$\phi_{ij}^{\min}, \phi_{ij}^{\max}$: minimum and maximum size factors when unit j is used for performing task i.

UC_j : set of potential conditions of unit j.

$I_{jkk'}$: set of tasks that take unit j from condition k to condition k'.

The campaign scheduling formulation is based on the following variables:

Binary Variables:

W_{ijt}^c = 1 if unit j starts processing task i at the beginning of the time period t in a cycle of campaign c; 0 otherwise.

$\bar{W}_{jkt}^c$ = 1 if unit j is idle in condition k over time period t in a cycle of campaign c; 0 otherwise.

Continuous Variables:

B_{ijt}^c = amount of material that starts undergoing task i at the beginning of the time period t in a cycle of campaign c.

S_{st}^c = amount of material stored in state s during time period t in a cycle of campaign c.

U_{ut}^c = amount of utility u used during time period t in a cycle of campaign c.

Π_{jt}^c = number of task batches which have been performed since the last cleaning operation in unit j at the end of time interval t in a cycle of campaign c.

In order to facilitate the description of the formulation, we use the cyclic operator, $\tau_c(t)$, defined by:

$$\tau_c(t) = \begin{cases} t & \text{if } t \geq 1 \\ \tau_c(t + T_c) & \text{if } t \leq 0 \end{cases}$$

where T_c is the cycle time for campaign c.

Below, we describe in detail all the appropriate constraints that need to be taken into account for the multiple campaign planning/scheduling problem.

3.3.1 Allocation Constraints

Unit allocation constraints state that, if a campaign exists, then every unit can perform at most one task at a time in a non pre-emptive fashion. This is written as:

$$\sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} W^c_{ij,\tau_c(t-\theta)} \leq X_c \quad \forall\, c,\, j,\, t = 1..T_c \tag{3.25}$$

If X_c equals one, then the above constraints lead to the constraints described by Shah *et al.* (1993a). On the other hand, if a campaign does not exist, then X_c is zero, and constraints (3.25) force all the variables in the double summation to zero.

The variables Y^c_{ij} characterising the utilisation of equipment items in each campaign can be defined through constraints of the form:

$$Y^c_{ij} \geq \sum_{t=\lambda p_i+1}^{\min(T_c,(\lambda+1)p_i)} W^c_{ijt} \quad \forall\, c,\, i,\, j \in K_i,\, \lambda = 0..\left\lfloor \frac{T_c}{p_i} \right\rfloor \tag{3.26}$$

The above constraints ensure that, if Y^c_{ij} is zero, then no instance of task i in unit j is allowed to occur during campaign c. On the other hand, it should be noted that *at most one* of the W^c_{ijt} variables in the summations on the right hand side can have a value of one at the same time. This is a consequence of the allocation constraints (3.25) and the fact that (3.26) divide the cycle time into segments not exceeding the processing time of task i.

If less tight constraints are acceptable, then (3.26) can be aggregated to:

$$\left\lfloor \frac{T_c}{p_i} \right\rfloor Y^c_{ij} \geq \sum_{t=1}^{T_c} \sum_{j \in K_i} W^c_{ijt} \quad \forall\, c,\, i,\, j \in K_i \tag{3.26'}$$

3.3.2 Batchsize Constraints

The amount of material that starts undergoing task i in unit j at time period t during campaign c is bounded by the minimum and maximum unit capacity.

$$\phi^{\min}_{ij} V_j W^c_{ijt} \leq B^c_{ijt} \leq \phi^{\max}_{ij} V_j W^c_{ijt} \quad \forall\, c,\, j,\, i \in I_j,\, t = 1..T_c \tag{3.27}$$

Note that the constant factors $\phi^{\min}_{ij}$ and $\phi^{\max}_{ij}$ ensure that the operating batchsize is bounded by the appropriate minimum and maximum capacities.

3.3.3 Material Balance Constraints within a Cycle

These constraints state that the amount of a state s held in storage over time interval t in campaign c equals the corresponding amount over interval $t-1$, plus the amount of material added to state s at the beginning of time period t, minus the amount of material removed from state s at the beginning of time period t. Material addition is a consequence of tasks i starting at time $t - p_{is}$, while material removal from state s is due to tasks starting at the beginning of time period t.

$$S_{st}^{c} = S_{s,t-1}^{c} + \sum_{i \in \bar{T}_s} \sum_{j \in K_i} \bar{\rho}_{is} B_{ij,\tau_c(t-p_{is})}^{c} - \sum_{i \in T_s} \sum_{j \in K_i} \rho_{is} B_{ijt}^{c} \quad \forall\, c,\, s,\, t = 1..T_c + 1 \qquad (3.28)$$

Note that in the second summation in the right hand side of (3.28), the cyclic operator $\tau_c(.)$ must be used as material added to a state may be produced by tasks starting either within the current cycle or during the previous one, spanning the two consecutive cycles.

Also, storage capacity constraints must be imposed on all states depending on the dedicated storage C_s available for that state:

$$0 \le S_{st}^{c} \le C_s \quad \forall\, c,\, s,\, t = 1..T_c + 1 \qquad (3.29)$$

It should be noted that that the initial amount S_{s0}^{c} of every state in each cycle is a variable to be determined by the optimisation algorithm.

3.3.4 Utility Availability Constraints

Many tasks require utilities such as steam and manpower. The utility demand may vary over the duration of the task, and is modelled using a fixed, batchsize-independent term and a variable batchsize-dependent term. The value of these terms may vary discretely (at each time interval) over the duration of the task. So, the total requirement of utility u over the time interval t in campaign c is given by:

$$U_{ut}^{c} = \sum_{i} \sum_{j \in K_i} \sum_{\theta=0}^{p_i - 1} \left(\alpha_{ui\theta} W_{ij,\tau_c(t-\theta)}^{c} + \beta_{ui\theta} B_{ij,\tau_c(t-\theta)}^{c} \right) \quad \forall\, c,\, u,\, t = 1..T_c \qquad (3.30)$$

Furthermore, the amount of utility u consumed over time period t for every campaign cannot exceed the maximum utility availability for this campaign, $U_{uc}^{\max}$:

$$0 \le U_{ut}^{c} \le U_{uc}^{\max} \quad \forall\, c,\, u,\, t = 1..T_c \qquad (3.31)$$

3.3.5 Cleaning Constraints

The need for cleaning the processing equipment between different uses is very common in practice in batch processes, and especially when quality-critical products such as pharmaceuticals and foods are involved. There are two types of cleaning: sequence-dependent and frequency-dependent cleaning.

3.3.5.1 Sequence-dependent Cleaning Constraints

We adopt the approach proposed by Pantelides (1994) which considers equipment in different states of cleanliness as different resources, with processing and cleaning tasks transforming one type of resource to another.

According to our definition, $\bar{W}^c_{jkt}$ is 1 only if unit j is *idle* in condition k. The condition of an idle unit over period t in campaign c depends on

- its condition during the previous time period;
- whether any tasks leading to that condition have started $t - p_i$ time intervals before; and
- whether any tasks starting from that condition start at time t within the same campaign.

Mathematically, this can be expressed as:

$$\bar{W}^c_{jkt} = \bar{W}^c_{jk,t-1} + \sum_{k'} \sum_{i \in I_{jk'k}} W^c_{ij,\tau_c(t-p_i)} - \sum_{k'} \sum_{i \in I_{jkk'}} W^c_{ijt} \quad \forall\, c,\, j,\, k \in UC_j,\, t = 1..T_c \quad (3.32)$$

It should be noted that for $t = 1$ the above constraint involves the variable $\bar{W}^c_{jk0}$ on the right hand side, but due to the cyclic nature of the schedule this is identical to $\bar{W}^c_{jkT_c}$:

$$\bar{W}^c_{jk0} = \bar{W}^c_{jkT_c} \quad \forall\, c,\, j \quad (3.33)$$

Clearly the $\bar{W}^c_{jkt}$ variables can be treated as continuous ones due to the nature of the two double summations in (3.32) which can take only the values 1 or 0.

Constraint (3.32) can be viewed as a "number balance" ensuring that the number of units j that are idle or in use is conserved from one time period to the next. However, it does not actually define this number; we therefore need to fix it at least once during the cycle, say for $t = 1$:

$$\sum_{k \in UC_j} \bar{W}^c_{jk1} + \sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} W^c_{ij,\tau_c(1-\theta)} = X_c \quad \forall\, c,\, j \quad (3.34)$$

where the first summation on the left hand side represents unit j being idle in different conditions over time interval $t = 1$ while the second one corresponds to unit j being used or cleaned (*cf.* the allocation constraints (3.25)). Clearly the two summations must add up to 1 for any active campaign ($X_c = 1$). For inactive campaigns, (3.34) together with (3.32) and (3.25) ensure that all $\bar{W}^c_{jkt}$ are forced to zero.

Constraints (3.32) are valid provided that each task i that can take place in unit j is allowed to do so only if the unit is in given unique condition k. This is not always true in practice: for instance, it may be possible to perform the same processing task both in a clean vessel and in a partially dirty one; or it may be possible to apply the same cleaning operation to a vessel being in a number of different "dirty" conditions. In such cases, we split task i into a number of tasks all effecting the same material transformations but taking place in the unit being in different initial conditions.

The STN modification is shown in figure 3.6.

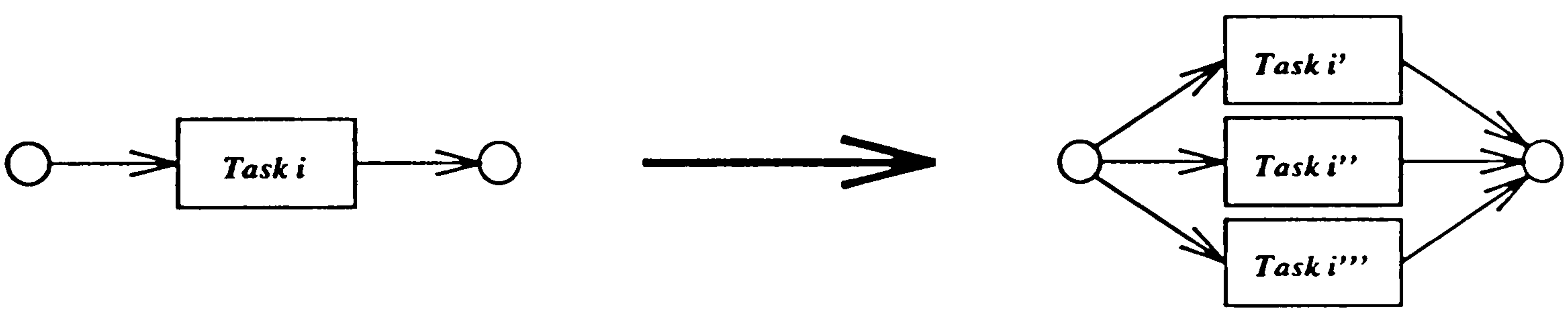


Figure 3.6: Task Splitting

Although we introduce additional processing tasks, there is no need to generate more variables in the mathematical formulation of the problem other than the corresponding binary ones, *i.e.* $W^c_{i'jt}$, $W^c_{i''jt}$, and $W^c_{i'''jt}$. These variables will be used only within constraints (3.32) replacing appropriately the W^c_{ijt} variables, while within the rest of the mathematical model (*e.g.* in the allocation constraints (3.25)) only the W^c_{ijt} variables will be used. Finally, an additional set of constraints is introduced to relate the new allocation variables to the original one:

$$W^c_{ijt} = W^c_{i'jt} + W^c_{i''jt} + W^c_{i'''jt} \qquad \forall\, c, j, t = 1..T_c \tag{3.35}$$

Evidence illustrating the relative effectiveness of this type of formulation for sequence-dependent cleaning over earlier ones (*e.g.* Crooks, 1992) has recently been provided by Barbosa-Póvoa (1994).

3.3.5.2 Frequency-dependent Cleaning Constraints

In this case, the cleaning requirements depend on the frequency with which a particular equipment item is used. Usually, this type of cleaning is used in order to prevent the build-up of impurities in processing items in processes in which the quality of final products is of great importance (*e.g.* polymer manufacture).

A unit subject to such cleaning requirements is not allowed to process more tham m_j *successive* batches, unless a cleaning task i^* takes place.

A set of constraints for frequency-dependent cleaning in short-term scheduling formulations were first proposed by Kondili *et al.* (1993), and similar constraints apply for the cyclic case. In fact, as noted by Barbosa-Póvoa (1994), only one group of the Kondili *et al.* constraints are really necessary:

$$\Pi^c_{jt} \geq \Pi^c_{j,\tau_c(t-1)} + \sum_{i \in I_j - \{i^*\}} W^c_{ijt} - m_j W^c_{i^*jt} \quad \forall\ c, j, t = 1..T_c \tag{3.36}$$

Here Π^c_{jt} represents the number of batches processed by unit j since its last cleaning and is therefore bounded by:

$$0 \leq \Pi^c_{jt} \leq m_j \quad \forall\ c, j, t = 1..T_c \tag{3.37}$$

Note that every time a processing task i takes place, the value of Π^c_{jt} is augmented at least by one. On the other hand, when the cleaning task i^* starts ($W^c_{i^*jt} = 1$), the Π^c_{jt} variable is allowed to return to a value of zero.

In order to reduce the potential solution degeneracy by forcing as many of the Π^c_{jt} variables as possible to zero, small penalty terms of the form

$$-\varepsilon \sum_j \sum_c \sum_{t=1}^{T_c} \Pi^c_{jt} \tag{3.38}$$

can be added to the objective function, where ε a small penalty number.

Albeit an integer, Π^c_{jt} can be treated as a continuous variable since its integrality is ensured by constraints (3.36), the integrality of the W^c_{ijt} variables, and the penalty (3.38) which ensures that either $\Pi^c_{jt} = 0$ or (3.36) are active.

3.3.6 Semi-continuous Operation Constraints

Apart from continuous and batch tasks, it is very common in practice to have to deal with semi-continuous tasks which operate in a continuous manner over short time periods. In our formulation, the latter can be modelled approximately as batch tasks with processing time equal to 1 time interval with an equivalent equipment processing capacity of

(production rate) × (duration of time interval).

The above model may be insufficient if it is necessary to operate semi-continuous equipment uninterruptedly for a given minimum duration of time ("run-length"). Crooks (1992) describes in detail the relevant constraints for the short-term scheduling problem; their extension to the campaign case is straightforward:

$$\sum_{\theta=1}^{\tau_{ij}^{RL}-1} W_{ij,t+\theta}^{c} \geq (\tau_{ij}^{RL} - 1)\,(W_{ijt}^{c} - W_{ij,\tau_c(t-1)}^{c}) \quad \forall\, c,\, j,\, t = 1..T_c,\, i \in CT,\, j \in K_i \qquad (3.39)$$

where CT is the set of semi-continuous tasks, and τ_{ij}^{RL} is the minimum run-length for task i taking place on semi-continuous unit j.

The above constraints are active only if $W_{ijt}^{c} = 1$ and $W_{ij,\tau_c(t-1)}^{c} = 0$, *i.e.* if the semi-continuous task actually starts at time t following a period of idleness. In this case, all the W_{ijt}^{c} variables involved in the summation on the left hand side of (3.39) are forced to take the value of one, *i.e.* unit j is allocated to task i for at least τ_{RL} time intervals.

3.4 Mathematical Formulation — Objective Function

A typical objective function, Φ, is the maximisation of the combined production value over the horizon of interest, taking into account the value of the products, the cost of the feeds, the value of the material that is received and/or delivered at distinct time instances, the cost of the utilities, the inventory cost for all the different types of material, and the cost of cleaning.

We consider each of these terms separately below.

3.4.1 Value of Products and Cost of Raw Materials

As we have already seen, product deliveries and raw materials may occur at distinct points during or at the end of the time horizon, as well as continuously during the horizon. We must also account for the nett changes in the inventory of each material over the entire horizon.

Mathematically, we have terms of the form:

$$+\sum_s \left(v_s \left(BS_{sN}^{F'} - BS_{s1}^{I}\right) + \sum_\mu \xi_{s\mu} \left(D_{s\mu} - R_{s\mu}\right) + \sum_c \zeta_{sc}(d_{sc} - r_{sc}) n_c T_c \right) \tag{3.40}$$

The above summation is, in principle, taken over all materials s in the process. The first term in the summation multiplies the nett change $(BS_{sN}^{F'} - BS_{s1}^{I})$ in the inventory of material s over the horizon by a nominal unit value, v_s. For raw materials and saleable end products, this may well be a true market price, but the latter may be difficult to establish for intermediates. In this case, v_s simply reflects the desirability of material s and it may depend on the degree of processing (*i.e.* how far along the path from raw material(s) to end product(s) state s actually occurs). For unwanted byproducts, v_s may take a negative value reflecting the cost of disposal.

The second term in (3.40) represents the value of product deliveries and raw material receipts that take place at distinct points during the horizon. Here, $\xi_{s\mu}$ is the true price realised or paid on each occasion.

The final term corresponds to continuous receipts or deliveries. The true price involved in these transactions, ζ_{sc}, may be seasonally dependent and may therefore vary from campaign to campaign. In general, since ζ_{sc}, d_{rc}, r_{sc} and n_c can all be variables, the last term of (3.40) is highly nonlinear. On the other hand, it the prices and delivery and receipt rates for the various materials are *known* functions of time, the nett revenue to be realised from continuous deliveries and receipts can be calculated *a priori* and need not therefore be introduced in the objective function.

3.4.2 Utility Costs

The cost of utilities may be described mathematically as:

$$-\sum_u c_u \sum_c n_c \sum_{t=1}^{T_c} U_{ut}^c \tag{3.41}$$

where c_u is the unit cost of utility u. Note that the innermost summation corresponds to the total amount of utility used over a cycle in campaign c. This is multiplied by the number of cycles, n_c, to obtain the total amount used over the entire campaign.

The above constraints involve nonlinear terms. By replacing the number of cycles by constraints (3.3), we derive bilinear terms. Consequently, we introduce new continuous variables $\hat{U}^c_{iut}$ defined as:

$$\hat{U}^c_{iut} \equiv L_{ic} U^c_{ut} \quad \forall\, c, u, i = 0..r_c - 1, t = 1..T_c$$

and replacing (3.41) by

$$-\sum_u c_u \sum_c \sum_{i=0}^{r_c-1} 2^i \sum_{t=1}^{T_c} \hat{U}^c_{iut} \tag{3.42}$$

The $\hat{U}^c_{iut}$ are defined implicitly through the constraints:

$$U^c_{ut} - (1 - L_{ic}) U^{\max}_{uc} \leq \hat{U}^c_{iut} \quad \forall\, c, u, i = 0..r_c - 1, t = 1..T_c \tag{3.43}$$

Due to the cost of utility the optimisation algorithm will always force the $\hat{U}^c_{iut}$ to be as small as possible. So, for the case of $L_{ic} = 1$, constraints (3.43) will force $\hat{U}^c_{iut}$ to the value of U^c_{ut}, whilst if $L_{ic} = 0$ constraints (3.43) become inactive and the $\hat{U}^c_{iut}$ variables will be forced to their lower bounds (namely zero).

3.4.3 Cleaning Costs

The constraints of section 3.3.5 characterise completely the occurrence of cleaning tasks, and therefore their cost can be taken into account in the objective function:

$$-\sum_j \sum_{i^* \in I^*_j} c_{i^*j} \sum_c n_c \sum_{t=1}^{T_c} W^c_{i^*jt} \tag{3.44}$$

where I^*_j is the set of all different types of cleaning task that may be applied to unit j, and c_{i^*j} is the cost of cleaning task of type i^* applied to j. The innermost summation corresponds to the number of cleaning tasks of type i^* that take place in unit j over a cycle of campaign c. This is multiplied by n_c to obtain the total number over the entire campaign.

The nonlinear terms involved in the above constraints can be substituted by their linear equivalents after some straightforward modifications. Replacing the number of cycles, n_c, using constraints (3.3), we come up with cross-products of binary variables. We introduce new continuous variables $\hat{W}^c_{ii^*jt}$ defined as:

$$\hat{W}^c_{ii^*jt} \equiv L_{ic}\, W^c_{i^*jt} \quad \forall\, c,\, j,\, i^*,\, i = 0..r_c - 1,\, t = 1..T_c$$

which allows us to replace (3.44) by the linear expression:

$$-\sum_j \sum_{i^* \in I^*_j} c_{i^*j} \sum_c \sum_{i=0}^{r_c - 1} 2^i \sum_{t=1}^{T_c} \hat{W}^c_{ii^*jt} \tag{3.45}$$

Also, the following constraints must be added to the formulation:

$$W^c_{i^*jt} + L_{ic} - 1 \le \hat{W}^c_{ii^*jt} \quad \forall\, c,\, j,\, i^*,\, i = 0..r_c - 1,\, t = 1..T_c \tag{3.46}$$

Note that if $L_{ic} = 0$, then the optimisation will force the $\hat{W}^c_{ii^*jt}$ to zero. Otherwise, if $L_{ic} = 1$, the $\hat{W}^c_{ii^*jt}$ will be set equal to $W^c_{i^*jt}$.

3.4.4 Inter-campaign Change-over Costs

The inter-campaign change-overs needed for plant reconfiguration, maintenance and/or unit cleaning may not only imply loss of production time, but may also be associated with certain costs. If the cost of change-over between campaigns c and $c + 1$ is denoted by η_c, then the following term must be added to the objective function:

$$-\sum_{c=1}^{N-1} \eta_c \tag{3.47}$$

We assume that η_c is proportional to the difficulty of change-over as expressed by the similarity measure SM_c (*cf.* section 3.2.8). This can be written as:

$$\eta_c \ge \eta^* \,(X_{c+1} - SM_c) \quad \forall\, c = 1..N - 1 \tag{3.48}$$

We note that if campaign $c + 1$ is active, then the optimisation will force η_c to $\eta^*(1 - SM_c)$ where η^* is the cost of completely reconfiguring the plant. On the other hand, if $X_{c+1} = 0$, then (3.48) is not active, and the optimisation will force η_c to its lower bound of zero.

3.4.5 Inventory Costs

To determine the cost of holding inventory we need to consider the variation of the amount of each material being kept in storage during the time horizon of interest. Consider, for instance, a typical product. From a macroscopic point of view, the inventory level variation will be of a piecewise linear form as shown in figure 3.7:

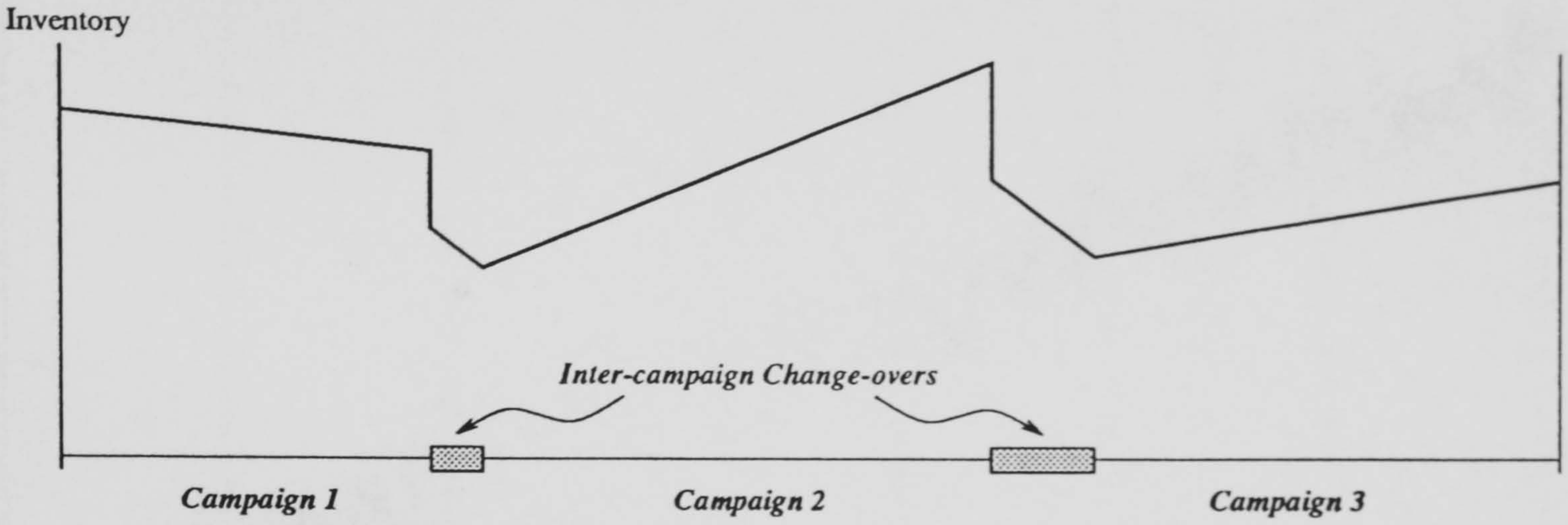


Figure 3.7: Macroscopic View of Inventory Level Variation

where a positive slope indicates campaign(s) during which the product is actually being produced while negative slope regions correspond to campaigns where either no such production takes place or, perhaps, the rate of continuous consumption exceeds the rate of production. Vertical drops in inventory levels indicate point demands or receipts for the material being satisfied.

However, the microscopic view of the inventory variation within each campaign can be more complex. Consider, for instance, a cycle in campaign 2 during which three batches of product are produced. Then the inventory profile over the cycle may be of the form shown in figure 3.8:

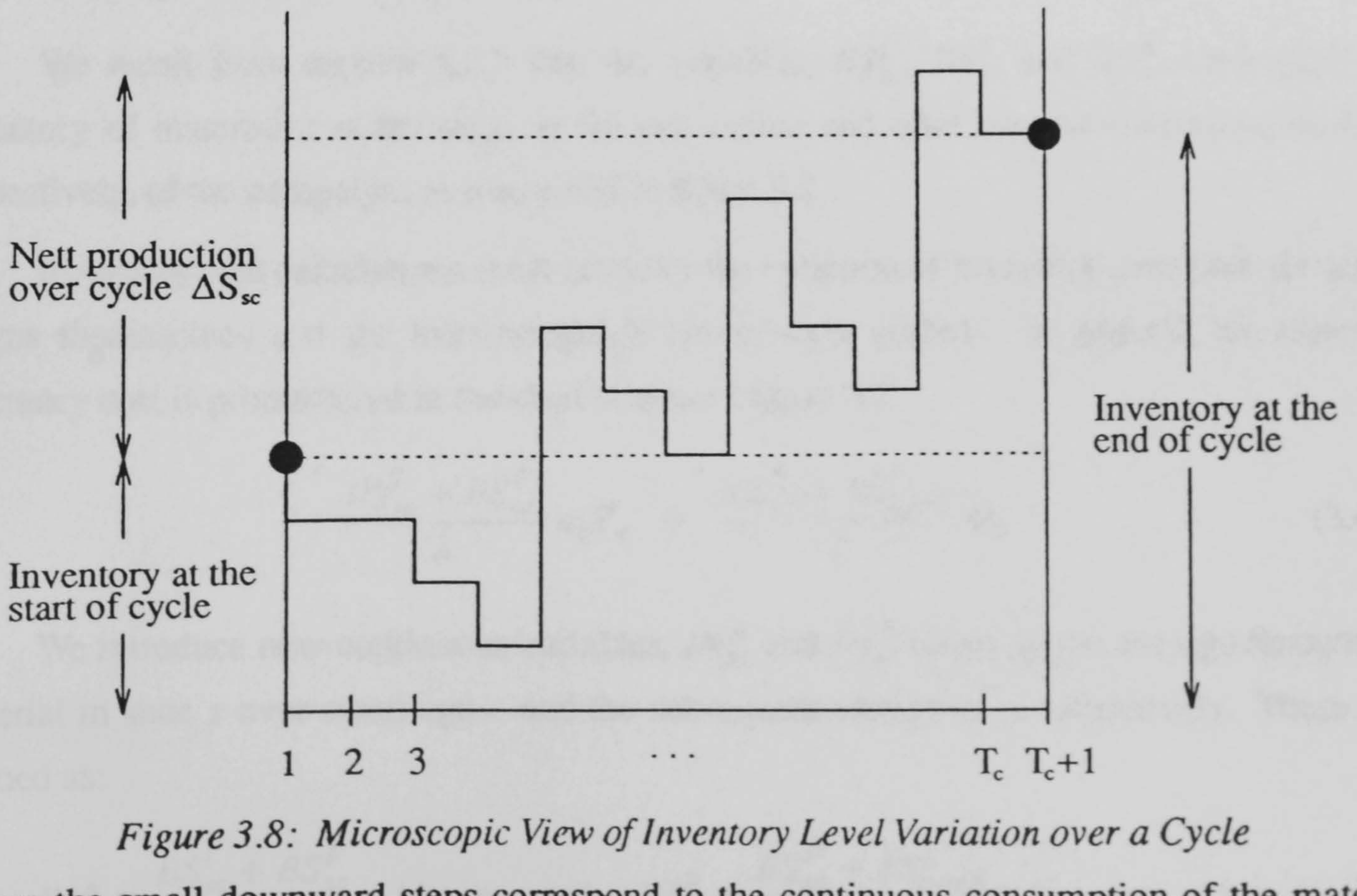


Figure 3.8: Microscopic View of Inventory Level Variation over a Cycle

where the small downward steps correspond to the continuous consumption of the material while the three large upward steps correspond to the three batches being produced.

The inventory profile over the entire campaign 2 can be obtained by juxtaposing a number of the cycle profiles as illustrated in figure (3.9):

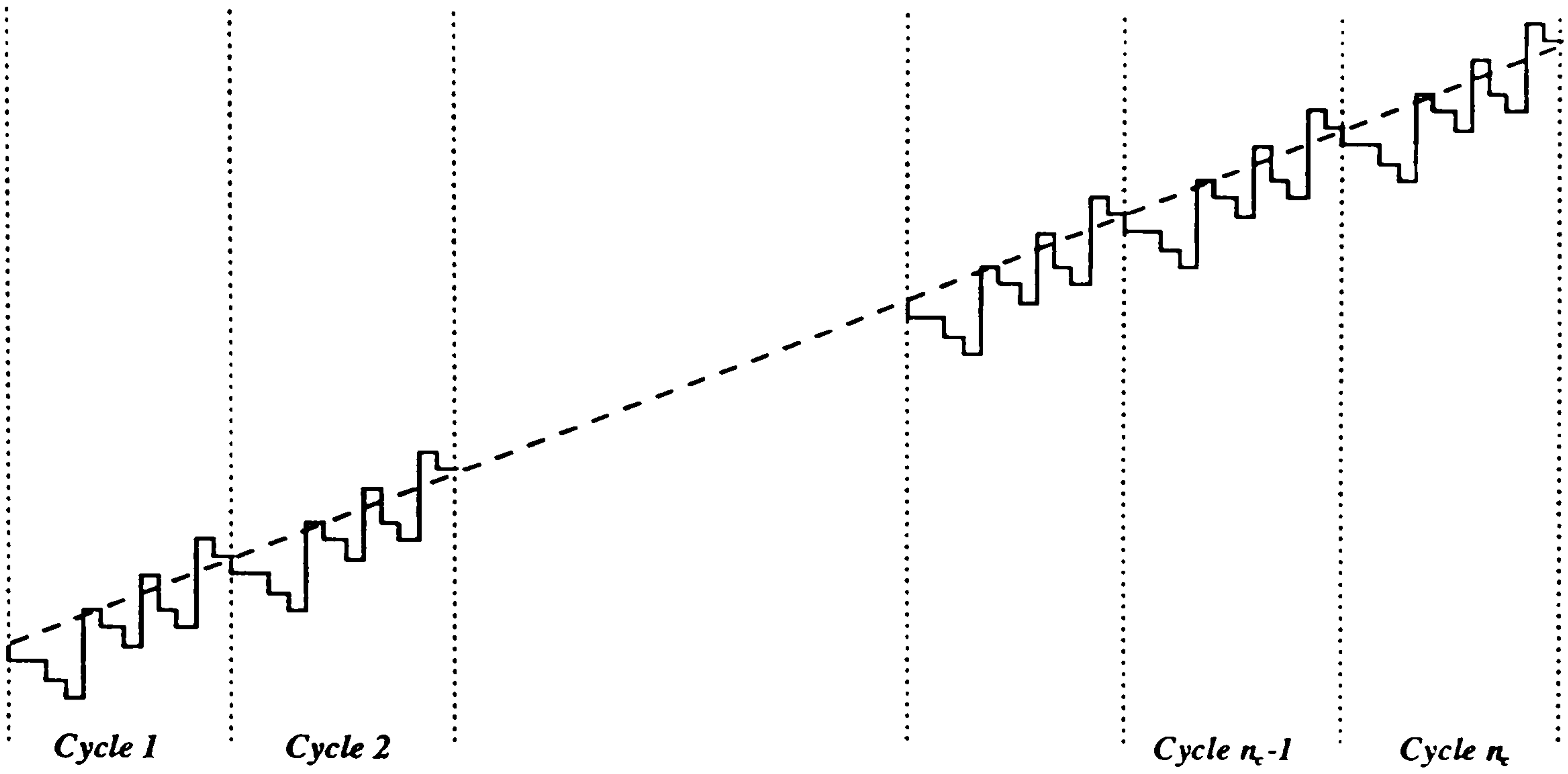


Figure 3.9: Precise Inventory Profile over Campaign

The precise inventory cost for campaign 2 is proportional to the area under the solid line shown above. However, provided that $n_c \gg 1$, a reasonable approximation can be obtained by assuming a linear inventory profile over each cycle, and therefore the entire campaign. On the other hand, if the number of cycles is small, the linear approximation will not be so good *but* in such a case the contribution of the inventory cost to the overall objective function will probably be negligible anyway.

We recall from section 3.2.5 that the variables, BS^I_{sc}, BS^F_{sc} and $BS^{F'}_{sc}$ correspond to inventory of material s at the start, at the end *before* and *after* discrete deliveries/receipts, respectively, of the campaign, as illustrated in figure 3.2.

Inventory cost calculations must consider the variation of inventory over both the campaigns themeselves and the inter-campaign change-over periods. In general, the material inventory cost is proportional to the shaded area of figure 3.2:

$$\frac{BS^I_{sc} + BS^F_{sc}}{2} n_c T_c + \frac{BS^{F'}_{sc} + BS^I_{s,c+1}}{2} \Theta_c \tag{3.49}$$

We introduce new continuous variables, IN^A_{sc} and IN^B_{sc} denoting the average amount of material in state s over campaign c and the subsequent change-over respectively. These are defined as:

$$IN^A_{sc} \equiv \frac{BS^I_{sc} + BS^F_{sc}}{2} \quad \forall\, c, s \quad \text{and} \quad IN^B_{sc} \equiv \frac{BS^{F'}_{sc} + BS^I_{s,c+1}}{2} \quad \forall\, c = 1..N-1, s$$

The relevant inventory cost can then be expressed as:

$$- \sum_s \alpha_s \sum_c \left(n_c T_c IN^A_{sc} + \Theta_c IN^B_{sc} \right) \tag{3.50}$$

where α_s reflects the opportunity cost of keeping the material in storage rather than selling it and investing the money realised. This constant takes into consideration the annual interest rate IR, and the value of the current state v_s, *i.e.*

$$\alpha_s = \frac{IR \times v_s}{365 \times 24}$$

with typical units of $£/kg.hr$.

Once again, we deal with bilinear terms $n_c IN^A_{sc}$ by replacing n_c by (3.3) and introducing new continuous variables IN^A_{ics} defined as:

$$IN^A_{ics} \equiv L_{ic} \frac{BS^I_{sc} + BS^F_{sc}}{2} \quad \forall\, c,\, s,\, i = 0..r_c - 1$$

along with the additional constraints:

$$\frac{BS^I_{sc} + BS^F_{sc}}{2} - (1 - L_{ic})\, M \le IN^A_{ics} \quad \forall\, c,\, s,\, i = 0..r_c - 1 \tag{3.51}$$

where M is an upper bound on $\frac{BS^I_{sc} + BS^F_{sc}}{2}$. This may take the value of the dedicated storage available for state s, C_s, or be based on some estimate of the production capacity.

The bilinear terms $IN^B_{sc}\Theta_c$ can either be ignored due to the relatively small duration of the inter-campaign change-overs, or be included by transforming them into linear forms. From (3.21) and (3.22) we obtain:

$$IN^B_{sc}\Theta_c \ge IN^B_{sc}\, X_{c+1}\, \Theta^F + IN^B_{sc} \left(X_{c+1} - \frac{\sum_i \lambda_{ic}}{NT} \right) \Theta^V \quad \forall\, c = 1..N-1,\, s \tag{3.52}$$

or

$$IN^B_{sc}\Theta_c \ge IN^B_{sc}\, X_{c+1}\, \Theta^F + IN^B_{sc} \frac{\sum_i (X_{c+1} - \lambda_{ic})}{NT} \Theta^V \quad \forall\, c = 1..N-1,\, s \tag{3.52'}$$

To linearise (3.52'), the IN^B_{sc} variables are replaced by two new continuous ones, IN^{B1}_{ics} and IN^{B2}_{cs}, defined as:

$$IN^{B1}_{ics} \equiv (X_{c+1} - \lambda_{ic}) \frac{BS^{F'}_{sc} + BS^I_{s,c+1}}{2} \quad \forall\, i,\, c = 1..N-1,\, s$$

and

$$IN_{cs}^{B2} \equiv X_{c+1} \frac{BS_{sc}^{F'} + BS_{s,c+1}^{I}}{2} \quad \forall\, c = 1..N-1,\, s,$$

and determined by the following constraints:

$$\frac{BS_{sc}^{F'} + BS_{s,c+1}^{I}}{2} - (X_{c+1} - \lambda_{ic})\, C_s \leq IN_{ics}^{B1} \quad \forall\, i,\, c = 1..N-1,\, s \tag{3.53}$$

and

$$\frac{BS_{sc}^{F'} + BS_{s,c+1}^{I}}{2} - (1 - X_{c+1})\, C_s \leq IN_{cs}^{B2} \quad \forall\, c = 1..N-1,\, s \tag{3.54}$$

Following the above linearisations, the inventory cost can be written as:

$$-\sum_s \alpha_s \sum_c \left(T_c \sum_{i=0}^{r_c-1} 2^i IN_{ics}^{A} + \frac{\Theta^V}{NT} \sum_i IN_{ics}^{B1} + \Theta^F IN_{cs}^{B2} \right) \tag{3.55}$$

3.5 Summary of Formulation

The entire formulation presented in this chapter is outlined below. Terms shown in bold can only be handled under certain simplifying — but very commonly valid — assumptions.

$$\Phi \equiv \max \quad \sum_{s}\left(v_s\,(BS_{sN}^{F'} - BS_{s1}^{I}) + \sum_{\mu} \xi_{s\mu}\,(D_{s\mu} - R_{s\mu}) + \sum_{\mathbf{c}} \zeta_{sc}(\mathbf{d}_{sc} - \mathbf{r}_{sc})\mathbf{n}_{c}\mathbf{T}_{c} \right)$$

$$- \sum_{u} c_u \sum_{c} \sum_{i=0}^{r_c-1} 2^i \sum_{t=1}^{T_c} \hat{U}_{iut}^{c} - \sum_{j} \sum_{i^* \in I_j^*} c_{i^*j} \sum_{c} \sum_{i=0}^{r_{c-1}} 2^i \sum_{t=1}^{T_c} \hat{W}_{ii^*jt}^{c}$$

$$- \sum_{c=1}^{N-1} \eta_c - \sum_{s} \alpha_s \sum_{c} \left(T_c \sum_{i=0}^{r_c-1} 2^i IN_{ics}^{A} + \frac{\Theta^V}{NT} \sum_{i} IN_{ics}^{B1} + \Theta^F IN_{cs}^{B2} \right)$$

s.t.

Campaign Planning Constraints

Campaign Timing Constraints

$$D^{\min} X_c \le n_c T_c \le D^{\max} X_c \quad \forall\, c$$

$$t_{c+1} \ge t_c + n_c T_c + \Theta_c \quad \forall\, c = 1..N-1$$

$$t_N + n_N T_N \le H$$

$$n_c = \sum_{i=0}^{r_c-1} 2^i\, L_{ic} \quad \forall\, c$$

Campaign Activity Constraints

$$X_{c+1} \le X_c \quad \forall\, c = 1..N-1$$

$$Y_i^c \le X_c \quad \forall\, c, i$$

$$Y_{ij}^c \le Y_i^c \quad \forall\, c, i, j \in K_i$$

Operability and Safety Constraints

$$\sum_{i \in \Xi} Y_i^c \le X_c \quad \forall\, c$$

$$\sum_i \psi_i Y_i^c \leq \Psi X_c \quad \forall\, c$$

$$\sum_{i \in I_j} Y_{ij}^c \leq n_j X_c \quad \forall\, c, j$$

Interior Point Event Constraints

$$\sum_c \Lambda_{\mu c} = 1 \quad \forall\, \mu$$

$$\sum_\mu \Lambda_{\mu c} \leq X_c \quad \forall\, c$$

$$H(1 - \sum_\mu \Lambda_{\mu c}) \geq t_c + n_c T_c - \sum_\mu \tau_\mu \Lambda_{\mu c} \geq 0 \quad \forall\, c$$

Material Balance Constraints over a Campaign

$$BS_{sc}^F = BS_{sc}^I + \sum_{i=0}^{r_c - 1} 2^i\, \hat{S}_{ics} \quad \forall\, c, s$$

$$BS_{sc}^{F'} = BS_{sc}^F + \sum_\mu \Lambda_{\mu c}(R_{s\mu} - D_{s\mu}) \quad \forall\, c, s$$

$$BS_{s,c+1}^I = BS_{sc}^{F'} + (\mathbf{r}_{s,c+1} - \mathbf{d}_{s,c+1})\Theta_c \quad \forall\, c, s$$

$$L_{ic}\left(\Delta S_{sc}^{\min} + T_c(r_{sc}^{\min} - d_{sc}^{\max})\right) \leq \hat{S}_{ics} \leq L_{ic}\left(\Delta S_{sc}^{\max} + T_c(r_{sc}^{\max} - d_{sc}^{\min})\right) \quad \forall\, c, s, i = 0..r_c - 1$$

$$(1 - L_{ic})\left(\Delta S_{sc}^{\min} + T_c(r_{sc}^{\min} - d_{sc}^{\max})\right) \leq \Delta S_{sc} + T_c(r_{sc} - d_{sc}) - \hat{S}_{ics} \leq (1 - L_{ic})\left(\Delta S_{sc}^{\max} + T_c(r_{sc}^{\max} - d_{sc}^{\min})\right)$$

$$\forall\, c, s, i = 0..r_c - 1$$

Storage Capacity and Inventory Constraints

$$0 \leq BS_{sc}^I \leq C_s\,; \quad 0 \leq BS_{sc}^F \leq C_s\,; \quad 0 \leq BS_{sc}^{F'} \leq C_s \quad \forall\, c, s$$

$$\sum_{s \in S_w} \omega_{ws} BS_{sc}^I \leq C_w\,; \quad \sum_{s \in S_w} \omega_{ws} BS_{sc}^F \leq C_w\,; \quad \sum_{s \in S_w} \omega_{ws} BS_{sc}^{F'} \leq C_w \quad \forall\, c, w$$

$$BS_s^{\min} \leq BS_{sc}^I \leq BS_s^{\max}\,; \quad BS_s^{\min} \leq BS_{sc}^F \leq BS_s^{\max}\,; BS_s^{\min} \leq BS_{sc}^{F'} \leq BS_s^{\max} \quad \forall\, c, s$$

Utility Availability Variations

$$U_{u,c+1}^{\max} = U_{uc}^{\max} + \sum_{\mu} \Lambda_{\mu c}\, \Delta U_{u\mu} \quad \forall\, c, u$$

Product Demand and Raw Material Supply Variations

$$r_{s,c+1}^{\max} = r_{sc}^{\max} + \sum_{\mu} \Lambda_{\mu c} \Delta r_{s\mu}^{\max} \quad \forall\, c, s$$

$$r_{s,c+1}^{\min} = r_{sc}^{\min} + \sum_{\mu} \Lambda_{\mu c} \Delta r_{s\mu}^{\min} \quad \forall\, c, s$$

$$d_{s,c+1}^{\max} = d_{sc}^{\max} + \sum_{\mu} \Lambda_{\mu c} \Delta d_{s\mu}^{\max} \quad \forall\, c, s$$

$$d_{s,c+1}^{\min} = d_{sc}^{\min} + \sum_{\mu} \Lambda_{\mu c} \Delta d_{s\mu}^{\min} \quad \forall\, c, s$$

$$r_{sc}^{\min} \le r_{sc} \le r_{sc}^{\max} \quad \forall\, c, s$$

$$d_{sc}^{\min} \le d_{sc} \le d_{sc}^{\max} \quad \forall\, c, s$$

Inter-campaign Change-over Constraints

$$SM_c = \frac{\sum_i \lambda_{ic}}{NT} \quad \forall\, c = 1..N-1$$

$$X_{c+1}\,(\Theta^F + \Theta^V) \ge \Theta_c \ge X_{c+1}\,\Theta^F + (X_{c+1} - SM_c)\,\Theta^V \quad \forall\, c = 1..N-1$$

$$\lambda_{ic} \ge 1 - Y_i^c - Y_i^{c+1} \quad \forall\, i, c = 1..N-1$$

$$\lambda_{ic} \ge Y_i^c + Y_i^{c+1} - 1 \quad \forall\, i, c = 1..N-1$$

$$\lambda_{ic} \le Y_i^c - Y_i^{c+1} + 1 \quad \forall\, i, c = 1..N-1$$

$$\lambda_{ic} \le Y_i^{c+1} - Y_i^c + 1 \quad \forall\, i, c = 1..N-1$$

Campaign Scheduling Constraints

Allocation Constraints

$$\sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} W^c_{ij,\tau_c(t-\theta)} \leq X_c \quad \forall\, c,\, j,\, t = 1..T_c$$

$$Y^c_{ij} \geq \sum_{t=\lambda p_i+1}^{\min(T_c,(\lambda+1)p_i)} W^c_{ijt} \quad \forall\, c,\, i,\, j \in K_i,\, \lambda = 0..\left\lfloor \frac{T_c}{p_i} \right\rfloor$$

Batchsize Constraints

$$\phi^{\min}_{ij} V_j W^c_{ijt} \leq B^c_{ijt} \leq \phi^{\max}_{ij} V_j W^c_{ijt} \quad \forall\, c,\, j,\, i \in I_j,\, t = 1..T_c$$

Material Balance Constraints within a Cycle

$$S^c_{st} = S^c_{s,t-1} + \sum_{i \in \bar{T}_s} \sum_{j \in K_i} \bar{\rho}_{is} B^c_{ij,\tau_c(t-p_{is})} - \sum_{i \in T_s} \sum_{j \in K_i} \rho_{is} B^c_{ijt} \quad \forall\, c,\, s,\, t = 1..T_c + 1$$

$$0 \leq S^c_{st} \leq C_s \quad \forall\, c,\, s,\, t = 1..T_c + 1$$

Utility Availability Constraints

$$U^c_{ut} = \sum_i \sum_{j \in K_i} \sum_{\theta=0}^{p_i-1} \left(\alpha_{ui\theta} W^c_{ij,\tau_c(t-\theta)} + \beta_{ui\theta} B^c_{ij,\tau_c(t-\theta)} \right) \quad \forall\, c,\, u,\, t = 1..T_c$$

$$0 \leq U^c_{ut} \leq U^{\max}_{uc} \quad \forall\, c,\, u,\, t = 1..T_c$$

Sequence-dependent Cleaning Constraints

$$\bar{W}^c_{jkt} = \bar{W}^c_{jk,t-1} + \sum_{k'} \sum_{i \in I_{jk'k}} W^c_{ij,\tau_c(t-p_i)} - \sum_{k'} \sum_{i \in I_{jkk'}} W^c_{ijt} \quad \forall\, c,\, j,\, k \in UC_j,\, t = 1..T_c$$

$$\bar{W}^c_{jk0} = \bar{W}^c_{jkT_c} \quad \forall\, c,\, j$$

$$\sum_{k \in UC_j} \bar{W}^c_{jk1} + \sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} W^c_{ij,\tau_c(1-\theta)} = X_c \quad \forall\, c,\, j$$

Frequency-dependent Cleaning Constraints

$$\Pi^c_{jt} \geq \Pi^c_{j,\tau_c(t-1)} + \sum_{i \in I_j - \{i^*\}} W^c_{ijt} - m_j W^c_{i^*jt} \quad \forall\, j,\, c,\, t = 1..T_c$$

$$0 \le \Pi_{jt}^{c} \le m_j \quad \forall\, c,\, j,\, t = 1..T_c$$

Semi-Continuous Operation Constraints

$$\sum_{\theta=1}^{\tau_{ij}^{RL}-1} W_{ij,t+\theta}^{c} \ge (\tau_{ij}^{RL} - 1)\,(W_{ijt}^{c} - W_{ij,\tau_c(t-1)}^{c}) \quad \forall\, c,\, j,\, t = 1..T_c,\, i \in CT,\, j \in K_i$$

Constraints Defining Objective Function Terms

$$U_{ut}^{c} - (1 - L_{ic})U_{uc}^{\max} \le \hat{U}_{iut}^{c} \quad \forall\, c,\, u,\, i = 0..r_c - 1,\, t = 1..T_c$$

$$W_{i*jt}^{c} + L_{ic} - 1 \le \hat{W}_{ii*jt}^{c} \quad \forall\, c,\, j,\, i*,\, i = 0..r_c - 1,\, t = 1..T_c$$

$$\frac{BS_{sc}^{I} + BS_{sc}^{F}}{2} - (1 - L_{ic})\, M \le IN_{ics}^{A} \quad \forall\, c,\, s,\, i = 0..r_c - 1$$

$$\frac{BS_{sc}^{F'} + BS_{s,c+1}^{I}}{2} - (X_{c+1} - \lambda_{ic})\, C_s \le IN_{ics}^{B1} \quad \forall\, i,\, c = 1..N-1,\, s$$

$$\frac{BS_{sc}^{F'} + BS_{s,c+1}^{I}}{2} - (1 - X_{c+1})\, C_s \le IN_{cs}^{B2} \quad \forall\, c = 1..N-1,\, s$$

3.6 Conclusions

This chapter presented a very general formulation for the multiple campaign planning problem for multipurpose batch plants. Many general features of batch processing such as material recycles, shared intermediates, mixed storage policies, limited resources, general operating mode of parallel units *etc.* are taken into account.

Furthermore, no *a priori* restriction is imposed on the possible campaign structure. The latter is determined automatically by the formulation taking account of all relevant factors such as:

- the availability of sufficient storage capacity to store large amounts of intermediates produced in one campaign for consumption over a later one;
- the cost of holding inventories;
- the product demand and raw material availability patterns; and
- the relative durations and costs of cleaning equipment within a campaign and inter-campaign plant reconfigurations.

Thus some of the key limitations of earlier approaches have been overcome.

The problem has been formulated as a MILP model, taking into account both planning and scheduling aspects simultaneously. The idea of having cyclic schedules within each campaign constitutes a convenient way for formulating our problem in a fashion similar to the one described by Shah *et al.* (1993a).

Of course, the simultaneous consideration of all decisions results in a rather large mixed integer programming formulation that can be an order of magnitude larger than similar formulations for short-term (Kondili *et al.*, 1993) and periodic (Shah *et al.*, 1993a) scheduling problems. In the next chapter, we consider ways of reducing the size and the integrality gap of this formulation by exploiting its special characteristics while an efficient decomposition approach is described in chapter 5.

Chapter 4

A Single-level Solution of the Campaign Planning/Scheduling Problem

The previous chapter established a general mathematical formulation of the campaign planning/scheduling problem. This resulted in a large MILP problem. In this chapter we consider ways in which the structure of the MILP can be exploited to reduce its size and the associated integrality gap.

These techniques include:

- exploitation of coupling between tasks to derive *a priori* constraints on the campaign structure;
- reduction of degeneracy resulting from
    - multiple identical campaigns; and
    - multiple identical equipment items;
- improvement of efficiency by eliminating infeasibilities in campaign structure; and
- tightening of the MILP through generation of valid inequalities.

The above are considered in turn in sections 4.1-4.5 below. An example of the application of the algorithm is presented in section 4.6.

4.1 Exploitation of Task Coupling

A key set of decisions in the campaign planning/scheduling problem are the subsets of the overall process STN that are active in each individual campaign. These are characterised by the binary variables Y_i^c which take the value of 1 if task i is active in campaign c and 0 otherwise.

In this section, we consider how the characteristics of the problem may be used to establish *a priori* relations between these variables. In particular, we examine tasks which are coupled through *intermediate* states of material, for which no external supplies or deliveries to customers take place.

4.1.1 Tasks Coupled through States with No Dedicated Storage Capacity

Consider first a state s with no dedicated storage capacity. If this state is produced by a single task i and consumed by a single task i' (see figure 4.1),

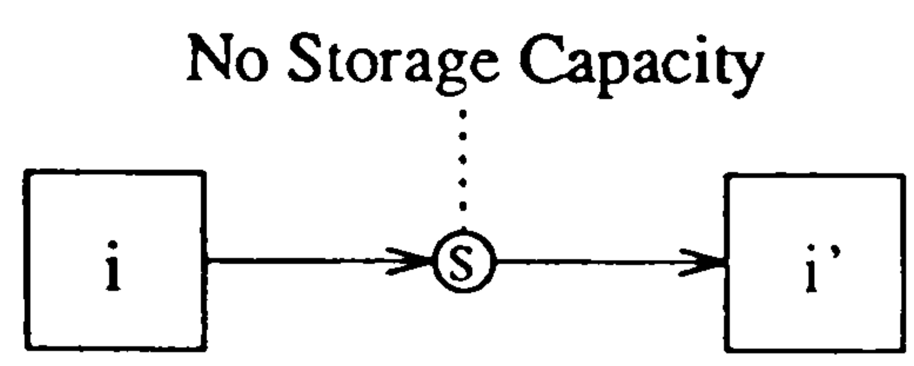


Figure 4.1: One Consumer — One Producer Tasks

then clearly i and i' must be both active or both inactive within any single campaign, leading to the following constraints:

$$Y_i^c = Y_{i'}^c \quad \forall\, c \tag{4.1a}$$

Thus, the Y variable need be defined for only one of the two tasks i, i'. Furthermore, (3.26) need also be written for one task only: if a W_{ijt}^c is 1 for a certain j and t, then there must be another $W_{i'j't'}^c$ that is also 1.

The above reasoning can be applied to trains of tasks with no intermediate storage between successive members of the train. The nett effect is automatically to group all of these tasks into a single "production stage", a concept used in some of the earlier formulations of the campaign planning/scheduling problem (see, for instance, Shah and Pantelides, 1991).

More generally, if a state s with no dedicated storage capacity is consumed by *only* one task, but is produced by several tasks (*i.e.* $card(T_s) = 1$, but $card(\bar{T}_s) > 1$) as shown in figure 4.2,

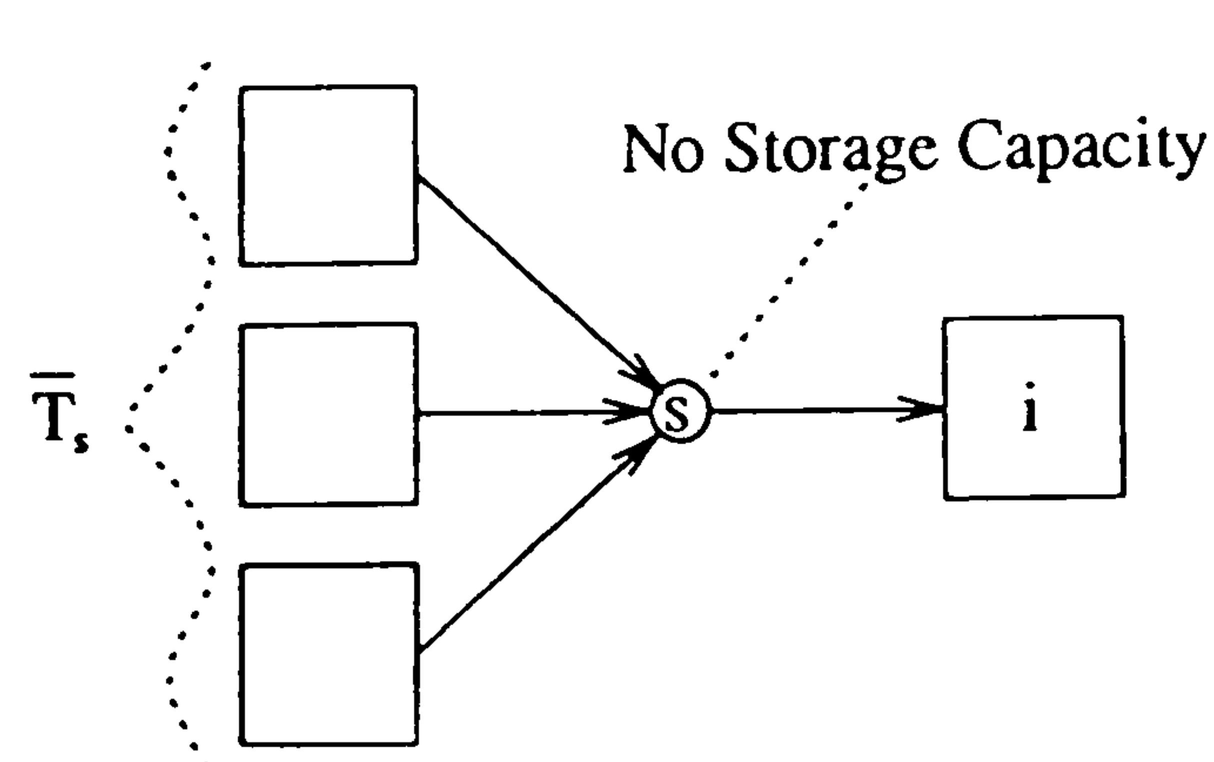


Figure 4.2: One Consumer — Many Producer Tasks

then we can write:

$$Y_i^c \geq Y_{i'}^c \quad \forall\, c,\ i' \in \bar{T}_s \tag{4.1b}$$

where $T_s = \{i\}$. In effect, if any task i' producing s is active in campaign c, then so must be the single task i consuming s. Conversely, if task i is involved in campaign c, then at least one task i' must be included in the same campaign as well:

$$\sum_{i' \in \overline{T}_s} Y_{i'}^c \geq Y_i^c \quad \forall\, c \tag{4.1c}$$

For the case in which $card(\overline{T}_s) = 1$ and $card(T_s) > 1$ as shown in figure 4.3,

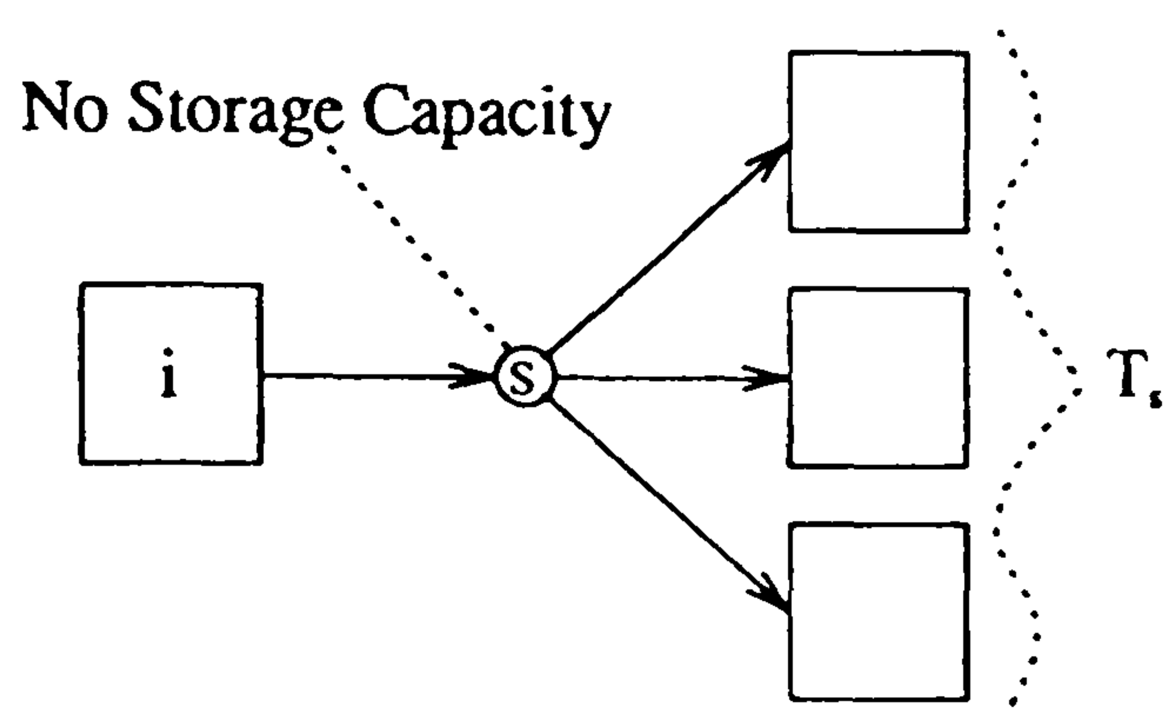


Figure 4.3: Many Consumer — One Producer Tasks

constraints similar to (4.1b and c) may be formulated:

$$Y_i^c \geq Y_{i'}^c \quad \forall\, c,\, i' \in T_s \tag{4.1d}$$

and

$$\sum_{i' \in T_s} Y_{i'}^c \geq Y_i^c \quad \forall\, c \tag{4.1e}$$

where $\overline{T}_s = \{i\}$. Note that if task i is *not* involved in campaign c, *i.e.* $Y_i^c = 0$, then all the immediate downstream tasks are also excluded from the same campaign. On the other hand, if task i *is* involved in campaign c, then (4.1c) force at least one of the preceding tasks to be performed within that same campaign.

4.1.2 Tasks Coupled through States with Limited Dedicated Storage Capacity

The above discussion was restricted to tasks coupled through states of material for which no dedicated storage was available. It is intuitively obvious that similar relationships may exist between tasks which are coupled through states with relatively small amounts of dedicated storage. The latter may be sufficient to permit short-term decoupling of the tasks within a single cycle of the same campaign, but not so large as to allow the tasks to operate in separate campaigns.

In general, in each campaign a subset of the tasks producing and consuming a state s will be active. We denote these subsets by $\overline{\Theta}_s$ and Θ_s ($\overline{\Theta}_s \subseteq \overline{T}_s$, $\Theta_s \subseteq T_s$) as illustrated in figure

4.4, where inactive tasks are shown as dashed boxes.

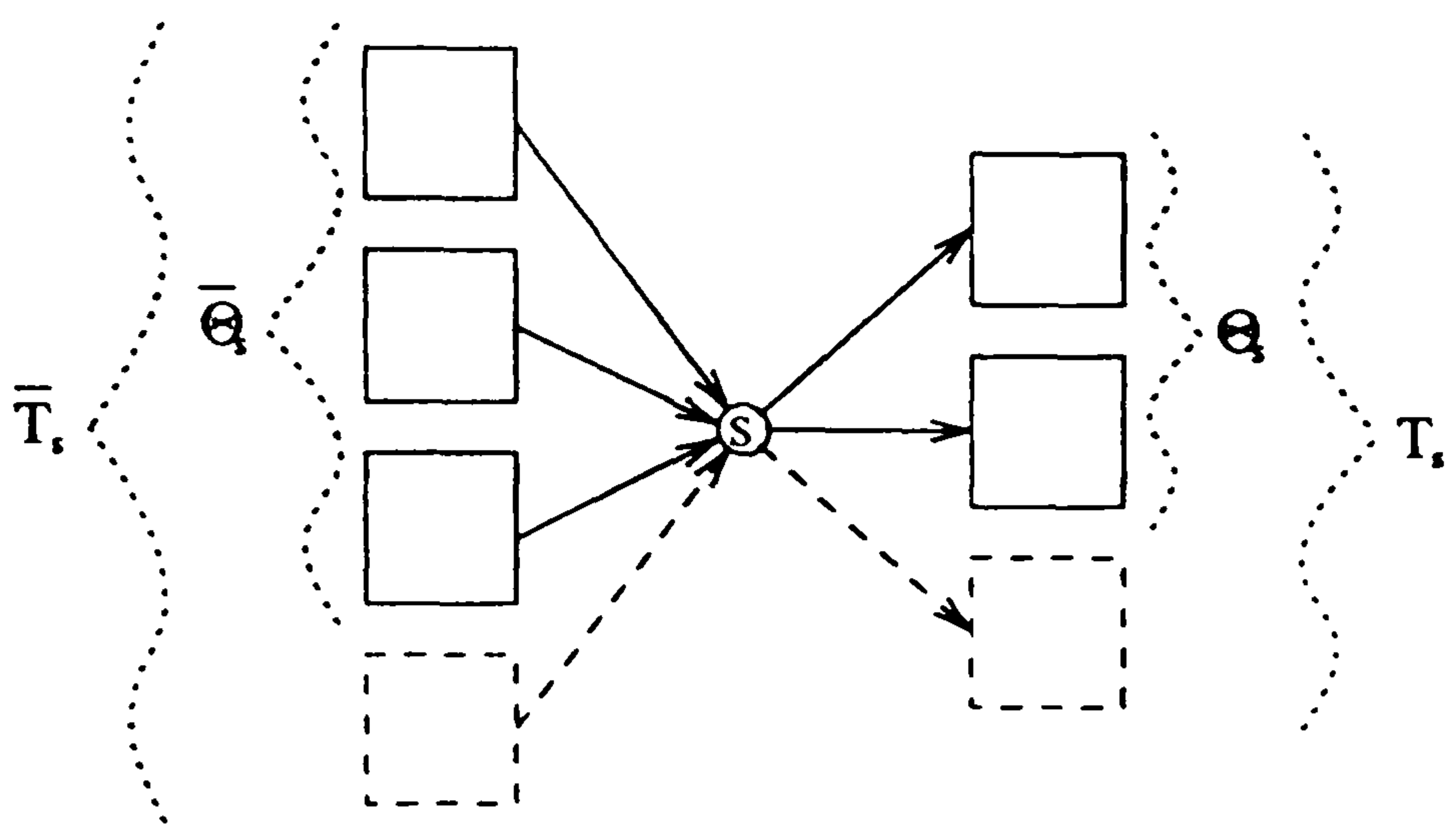


Figure 4.4: General State Representation

Furthermore, we denote the subset of equipment items that are actually used for carrying out task i during the campaign by $\bar{K}_i$ ($\bar{K}_i \subseteq K_i$). Note that, due to the multipurpose nature of the equipment, $\bar{K}_i$ and $\bar{K}_{i'}$ are not necessarily disjoint for two different tasks i and i'.

Obviously, any active task must take place at least once per cycle in each item of equipment used for it. Hence the minimum amount of material undergoing this task is:

$$\sum_{j \in \bar{K}_i} \phi_{ij}^{\min} V_j \tag{4.2}$$

obtained by setting all batchsizes at their minimum values.

On the hand, the maximum number of batches of material that can undergo a task i during a cycle of duration T_c is $\left\lfloor \frac{T_c}{p_i} \right\rfloor$. Thus by setting all batchsizes to their maximum value, we obtain the maximum amount of material that can undergo task i during a cycle:

$$\left\lfloor \frac{T_c}{p_i} \right\rfloor \sum_{j \in \bar{K}_i} \phi_{ij}^{\max} V_j \tag{4.3}$$

Now consider the nett accumulation of material in state s over a cycle, ΔS_{sc}. Its minimum value would occur if all tasks producing it ($i \in \bar{\Theta}_s$) produced the minimum possible amount of material whilst all tasks consuming it ($i \in \Theta_s$) processed the maximum possible amount. Hence the lower limit is:

$$\Delta S_{sc}^L \equiv \sum_{i \in \bar{\Theta}_s} \bar{\rho}_{is} \sum_{j \in \bar{K}_i} \phi_{ij}^{\min} V_j - \sum_{i \in \Theta_s} \rho_{is} \left\lfloor \frac{T_c}{p_i} \right\rfloor \sum_{j \in \bar{K}_i} \phi_{ij}^{\max} V_j \quad \forall\, c, s, \Theta_s, \bar{\Theta}_s, \bar{K}_i$$

On the other hand, the maximum possible nett accumulation of state s would occur if all tasks $i \in \bar{\Theta}_s$ processed as much material as possible and all tasks $i \in \Theta_s$ as little as possible:

$$\Delta S_{sc}^{U} \equiv \sum_{i \in \bar{\Theta}_s} \bar{\rho}_{is} \left\lceil \frac{T_c}{p_i} \right\rceil \sum_{j \in \tilde{K}_i} \phi_{ij}^{\max} V_j - \sum_{i \in \Theta_s} \rho_{is} \sum_{j \in \tilde{K}_i} \phi_{ij}^{\min} V_j \quad \forall\, c, s, \Theta_s, \bar{\Theta}_s, \tilde{K}_i$$

Having established bounds on the nett accumulation over a cycle, consider now the accumulation over an entire campaign of minimum duration $D_c^{\min}$. In particular, the minimum number of cycles is $\left\lfloor \frac{D_c^{\min}}{T_c} \right\rfloor$.

Consider first the case where $\Delta S_{sc}^{L} > 0$. Hence material must actually accumulate over the campaign. Then clearly the minimum nett accumulation over a campaign is:

$$\left\lfloor \frac{D_c^{\min}}{T_c} \right\rfloor \Delta S_{sc}^{L} \tag{4.4}$$

If this exceeds the available storage capacity C_s for this material, then the combination of active tasks Θ_s, $\bar{\Theta}_s$ and utilised equipment $\tilde{K}_i$ is illegal and can be excluded a priori.

Now consider the case where $\Delta S_{sc}^{U} < 0$, *i.e.* material is actually depleted during the campaign. The minimum nett depletion is given by:

$$\left\lfloor \frac{D_c^{\min}}{T_c} \right\rfloor (-\Delta S_{sc}^{U}) \tag{4.5}$$

The maximum possible initial amount of s that could be available at the start of a campaign is given by the dedicated storage capacity, C_s. If the latter is less than the minimum depletion (4.5), then again the combination of Θ_s, $\bar{\Theta}_s$, $\tilde{K}_i$ under consideration can be excluded *a priori*.

The elimination of a given $\{\Theta_s, \bar{\Theta}_s, \tilde{K}_i\}$ combination can be achieved through integer cuts on the binary variables Y_{ij}^{c} which describe whether or not equipment item j is used for task i during campaign c:

$$\sum_{i \in \Theta_s^{IN}} \sum_{j \in \tilde{K}_i} Y_{ij}^{c} - \sum_{i \in \Theta_s^{OUT}} \sum_{j \in \tilde{K}_i} Y_{ij}^{c} \le \left(\sum_{i \in \Theta_s^{IN}} card(\tilde{K}_i) \right) - 1 \quad \forall\, c, s \tag{4.6}$$

where $\Theta_s^{IN} = \Theta_s \cup \bar{\Theta}_s$ is the subset of tasks producing or consuming s that are active over campaign c, and $\Theta^{OUT} = (T_s \cup \bar{T}_s) - \Theta_s^{IN}$ its complement.

4.1.3 Elimination of W^c_{ijt} Variables

The main set of binary variables in our formulation is that of the equipment allocation variables W^c_{ijt}. Shah *et al.* (1993b) developed a specialised MILP solver which eliminated these key variables along with the related capacity constraints from the LP relaxation, and then reconstructed them from the solution obtained. This reduces the size of the LP examined, and therefore accelerates its solution. The idea is applicable to all STN-based discrete-time formulations, including the one presented here.

An additional variable elimination step could take place for the special but common case of zero-wait trains of tasks when only one item of processing equipment is suitable for each task in the train, *i.e.* $card(T_s) = card(\bar{T}_s) = 1$ and $card(K_i) = card(K_{i'}) = 1$ where $i \in \bar{T}_s$ and $i' \in T_s$.

This situation is depicted in figure 4.5.

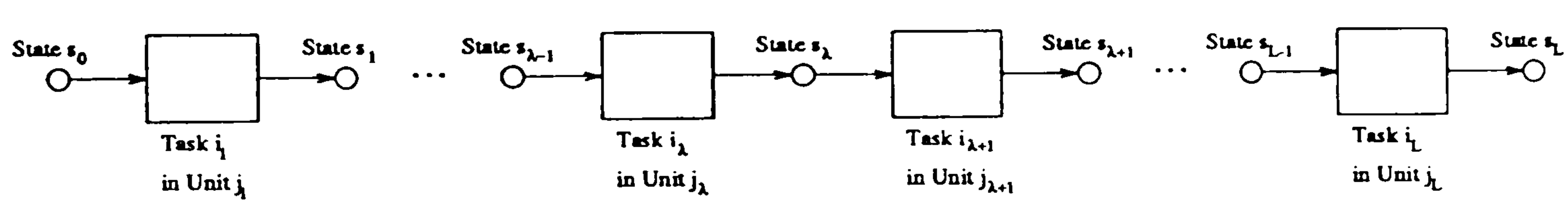


Figure 4.5: Zero-wait Train

For a train of length L, we can then write the constraints:

$$W^c_{i_{\lambda+1} j_{\lambda+1} t} = W^c_{i_\lambda j_\lambda, \tau_c(t-p_{i_\lambda})} \quad \forall\, c,\ t = 1..T_c,\ \lambda = 1..L-1 \tag{4.7a}$$

i.e. a task $i_{\lambda+1}$ may start in its corresponding unit at time t only if the previous task in the train, i_λ, started at time $t - p_{i_\lambda}$.

Through the mass balances, we can also relate the batchsizes of corresponding tasks:

$$\rho_{i_{\lambda+1} s_\lambda} B^c_{i_{\lambda+1} j_{\lambda+1} t} = \bar{\rho}_{i_\lambda s_\lambda} B^c_{i_\lambda j_\lambda, \tau_c(t-p_{i_\lambda})} \quad \forall\, c,\ t = 1..T_c,\ \lambda = 1..L-1 \tag{4.7b}$$

Constraints (4.7a, b) allow the elimination of all equipment allocation and batchsize variables for all but one task in the train (*e.g.* the first one). However, the batchsize constraints (3.27) on the first task must be amended to take account of any capacity limitations in any subsequent task:

$$\max_\lambda(\alpha_\lambda \phi^{min}_{i_\lambda j_\lambda} V_{j_\lambda}) \le B^c_{i_1 j_1 t} \le \min_\lambda(\alpha_\lambda \phi^{max}_{i_\lambda j_\lambda} V_{j_\lambda}) \tag{4.8}$$

where $\alpha_1 \equiv 1$ and $\alpha_{\lambda+1} = \alpha_\lambda \dfrac{\rho_{i_{\lambda+1} s_\lambda}}{\bar{\rho}_{i_\lambda s_\lambda}}$, $\lambda = 1..L-1$.

4.2 Elimination of Campaign Degeneracy

The formulation presented in chapter 3 assumes a certain maximum number of campaigns. It is obvious that if an optimal solution involves fewer campaigns than this maximum, then equivalent solutions may be obtained by artificially partitioning any one campaign into two or more identical successive campaigns, provided minimum campaign duration constraints are not violated. This solution degeneracy may be detrimental to the solution efficiency of the MILP.

One way of avoiding such degeneracy is to penalise the number of campaigns in the objective function through a term of the form:

$$-\varepsilon \sum_{c=1}^{N} X_c \tag{4.9}$$

when ε is a small number.

For problems that do not involve discrete point events (such as deliveries of products or changes in utility availability), there is no reason for two *consecutive* active campaigns to be identical. This possibility can therefore be excluded explicitly through the *a priori* introduction of constraints of the form:

$$\sum_i \lambda_{ic} \leq NT - X_c \quad \forall\, c \tag{4.10}$$

where λ_{ic} are the similarity variables introduced in section 3.2.8, and NT is the total number of processing tasks in the STN. We note that for two exactly identical campaigns, $\sum_i \lambda_{ic} = NT$, but this is prohibited by (4.10) if the campaigns are active (*i.e.* $X_c = 1$).

Furthermore, for problems not involving any discrete event or continuous deliveries/receipts of material during the horizon, no two active campaigns c and c' need be identical in the optimal solution even if they are non-consecutive. We therefore define similarity variables $\lambda_{icc'}$ for every appropriate pair of campaigns (c, $c > c'$) rather than just successive ones (c and $c + 1$) through the following constraints:

$$\lambda_{icc'} \geq 1 - Y_i^c - Y_i^{c'} \quad \forall\, i, c, c' > c \tag{4.11a}$$

$$\lambda_{icc'} \geq Y_i^c + Y_i^{c'} - 1 \quad \forall\, i, c, c' > c \tag{4.11b}$$

and introduce the cuts:

$$\sum_i \lambda_{icc'} \leq NT - X_c \quad \forall\, c, c' > c \tag{4.10'}$$

It is worth noting that constraints for avoiding campaign degeneracy have also been presented by Tsirukis *et al.* (1993). Their constraints are described briefly below (adapted to our notation). First, they introduce binary variables $\mu_{icc'}$ to be equal to one if task i is involved in *either* campaign c or c', and zero otherwise, along with the following constraints:

$$\mu_{icc'} \geq Y_i^c \quad \forall\, i,\, c,\, c' > c \tag{4.12a}$$

$$\mu_{icc'} \geq Y_i^{c'} \quad \forall\, i,\, c,\, c' > c \tag{4.12b}$$

and

$$\mu_{icc'} \leq Y_i^c + Y_i^{c'} \quad \forall\, i,\, c,\, c' > c \tag{4.12c}$$

Then, they introduce additional binary variables $\lambda_{icc'}$ to denote whether task i occurs in *both* campaigns c and c'. These are associated with the following constraints:

$$\lambda_{icc'} \leq Y_i^c \quad \forall\, i,\, c,\, c' > c \tag{4.13a}$$

$$\lambda_{icc'} \leq Y_i^{c'} \quad \forall\, i,\, c,\, c' > c \tag{4.13b}$$

and

$$\lambda_{icc'} \geq Y_i^c + Y_i^{c'} - 1 \quad \forall\, i,\, c,\, c' > c \tag{4.13c}$$

Every pair of campaigns c and c' is characterised by two distinct indices, $z_{cc'}$ and $r_{cc'}$ defined as:

$$z_{cc'} = \sum_{i=1}^{NT} 2^{i-1} \mu_{icc'} \quad \forall\, i,\, c,\, c' > c \tag{4.14a}$$

and

$$r_{cc'} = \sum_{i=1}^{NT} 2^{i-1} \lambda_{icc'} \quad \forall\, i,\, c,\, c' > c \tag{4.14b}$$

It should be noted that $z_{cc'} \geq r_{cc'}$, with $z_{cc'} = r_{cc'}$ only if every task occurs either in both c and c', or in neither. Hence identical campaigns can be avoided by:

$$z_{cc'} - r_{cc'} \geq 1 \quad \forall\, c,\, c' > c \tag{4.15}$$

However, as the binary variables, $\mu_{icc'}$ and $\lambda_{icc'}$, occur with large coefficients in constraints (4.14a, b), the measures for campaign differences are quite loose. In addition, the set of constraints (4.12-4.15) does not allow more than one empty campaign at the final plan, and is therefore inappropriate for our mathematical formulation. Finally, the number of variables and constraints involved in the formulation presented by Tsirukis *et al.* (1993) is larger than the one described earlier in this section.

4.3 Elimination of Equipment Degeneracy

Our current formulation is degenerate for plants involving multiple equipment items with exactly the same functionality and capacity characteristics. This *equipment degeneracy* stems from the fact that assignments of distinct but otherwise identical equipment result in the same optimal solution.

In order to avoid the above deficiency, we introduce the idea of an equipment *type* comprising all identical items. We can then replace the basic binary variables, W^c_{ijt} with new integer ones, N^c_{ijt}, defined as:

N^c_{ijt} : number of units of *type* j starting processing task i at the start of time period t.

The allocation constraints of the original formulation (3.25) should be replaced by:

$$\sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} N^c_{ij,\tau_c(t-\theta)} \le N^*_j X_c \quad \forall\, c, j, t \tag{4.16}$$

where N^*_j is the maximum available number of units of type j. Note that these constraints are written only once for each equipment *type* j and not for every equipment item separately.

In addition, the B^c_{ijt} variables now refer to the *aggregated* batch sizes (instead of individual ones), defined as the total amount of material that starts undergoing task i in items of type j at time t. The new batch size constraints are as follows:

$$N^c_{ijt}\phi^{\min}_{ij} V_j \le B^c_{ijt} \le N^c_{ijt}\phi^{\max} V_j \quad \forall\, c, j, i \in I_j, t \tag{4.17}$$

replacing (3.27).

The utility availability constraints (3.30) should also be modified accordingly:

$$U^c_{ut} = \sum_i \sum_{j \in K_i} \sum_{\theta=0}^{p_i-1} \left(\alpha_{ui\theta} N^c_{ij,\tau_c(t-\theta)} + \beta_{ui\theta} B^c_{ij,\tau_c(t-\theta)} \right) \quad \forall\, c, u, t \tag{4.18}$$

Finally, the sequence-dependent cleaning constraints can be written in terms of the new variables N^c_{ijt}. To do this, we replace $\bar{W}^c_{jkt}$ by $\bar{N}^c_{jkt}$ denoting the number of units of type j which are idle in condition k over time interval t in a cycle of campaign c. Constraints (3.32)-(3.34) are modified to:

$$\bar{N}^c_{jkt} = \bar{N}^c_{jk,t-1} + \sum_{k'} \sum_{i \in I_{jk'k}} N^c_{ij,\tau_c(t-p_i)} - \sum_{k'} \sum_{i \in I_{jkk'}} N^c_{ijt} \quad \forall\, c,\, j,\, k \in UC_j,\, t = 1..T_c \quad (4.19a)$$

$$\bar{N}^c_{jk0} = \bar{N}^c_{jkT_c} \quad \forall\, c,\, j \quad (4.19b)$$

and

$$\sum_{k \in UC_j} \bar{N}^c_{jk1} + \sum_{i \in I_j} \sum_{\theta=0}^{p_i-1} N^c_{ij,\tau_c(1-\theta)} = N^*_j X_c \quad \forall\, c,\, j \quad (4.19c)$$

Apart from eliminating degeneracy, the above formulation results in smaller numbers of variables and constraints, thus potentially requiring less computational time for their solution.

4.4 Elimination of Infeasibilities

During the solution of the MILP using branch-and-bound procedures, it is often the case that many of the relaxed LPs solved are found to be infeasible. It is clearly desirable to accelerate the MILP solution by avoiding partial assignments of the binary variables leading to such infeasible nodes on the search tree.

4.4.1 Scheduling Infeasibilities

The imposition of minimum run-length constraints (3.39) on the operation of semi-continuous units (*cf.* section 3.3.6), may cause many equipment allocations generated during the branch-and-bound search to be infeasible.

In particular this may arise if, by a certain stage in the search, a variable W^c_{ijt} has been fixed to 1. Now, if at this stage, variable $W^c_{ij,\tau_c(t-1)}$ is also fixed to zero, then this implies that the unit j actually starts processing task i at time t and should continue doing so for the next τ^{RL}_{ij} time intervals. However, this results in an infeasibility if any one of the variables $W^c_{ij,t+\theta}$, $\theta = 1..\tau^{RL}_{ij} - 1$ has already been fixed to zero at an earlier stage of the search.

Obviously such infeasible LPs may be avoided by checking for the above cases during the branch-and-bound procedure. Alternatively, constraints (3.39) could be disaggregated to:

$$W^c_{ij,t+\theta} \geq W^c_{ijt} - W^c_{ij,\tau_c(t-1)} \quad \forall\, c,\, t = 1..T_c,\, i \in CT,\, j \in K_i,\, \theta = 1..\tau^{RL}_{ij} - 1$$

This would eliminate any potential infeasibilities, as fixing $W^c_{ij,t+\theta}$ and $W^c_{ij,\tau_c(t-1)}$ to zero would immediately constrain W^c_{ijt} to zero and no branching on this variable would ever be attempted. However, it may also result in large number of constraints if $\tau^{RL}_{ij} >> 1$.

4.4.2 Planning Infeasibilities

For problems involving interior point events, event ordering constraints related to the time-increasing nature of these point events, *i.e.* $\tau_\mu < \tau_{\mu+1}$, can eliminate potential infeasible solutions. These constraints are of the form:

$$1 - \Lambda_{\mu c} \geq \sum_{c'=1}^{c} \Lambda_{\mu' c'} \quad \forall\ \mu, c, \mu' = \mu + 1..NDP \tag{4.20}$$

Note that if event μ occurs at the end of campaign c, *i.e.* $\Lambda_{\mu c} = 1$, then no subsequent event ($\mu' > \mu$) is allowed to occur *before* campaign c, and thus the $\Lambda_{\mu' c'}$ variables are forced to zero. On the other hand, if any $\Lambda_{\mu' c'} = 1$, then $\Lambda_{\mu c} = 0$.

The inclusion of a task within a certain campaign is often the source of infeasibility, if, for example, sufficient inputs for this task are not available at the start of the campaign. In particular, consider an intermediate state s that is neither received from external suppliers nor delivered to external clients. Then, if the initial inventory of this state is zero (*i.e.* $BS^I_{s1} = 0$), no task consuming s should be allowed to start processing in a certain campaign if its preceding tasks have not been included in this or previous campaigns. This can be written as:

$$\sum_{c'=1}^{c} \sum_{i' \in \overline{T}_s} Y^{c'}_{i'} \geq Y^c_i \quad \forall\ c, s \in SZ, i \in T_s \tag{4.21a}$$

where SZ is the set of states with zero initial inventory (*i.e.* $BS^I_{s1} = 0$).

If, now, we assume there exists an event μ involving a discrete non-zero external delivery of state s, $D^{\min}_{s\mu}$, which exceeds the initially available amount, BS^I_{s1}, then this delivery cannot take place before some material in this state is actually produced. This leads to:

$$\sum_{c'=1}^{c} \sum_{i' \in \overline{T}_s} Y^{c'}_{i'} \geq \sum_{c'=1}^{c} \Lambda_{\mu c'} \quad \forall\ \{\mu, s\} : D^{\min}_{s\mu} \geq BS^I_{s1}, c \tag{4.21b}$$

Constraints (4.21b) consider each discrete delivery separately, taking no account of the fact that there may be multiple events involving deliveries of the same state s, and hence the cumulative amount of the minimum amounts to be delivered may exceed BS^I_{s1} even if the individual amounts do not. Defining

$$\tilde{D}_{s\mu} \equiv \sum_{\mu'=1}^{\mu} D_{\mu's}^{\min}$$

to be the minimum cumulative delivery for state s up to and including event μ, we modify (4.21b) to:

$$\sum_{c'=1}^{c} \sum_{i' \in \bar{T}_s} Y_{i'}^{c'} \geq \sum_{c'=1}^{c} \Lambda_{\mu c'} \quad \forall \{\mu, s\} : \tilde{D}_{s\mu} \geq BS_{s1}^{l}, c \tag{4.21c}$$

Note that the above constraints do not need to be written for every event μ, but only for the smallest μ for which $\tilde{D}_{s\mu} \geq BS_{s1}^{l}$: the constraints referring to all subsequent events are looser than (4.21c) as $\sum_{c'=1}^{c} \Lambda_{\mu c'} \geq \sum_{c'=1}^{c} \Lambda_{\mu+1,c'}$.

Finally, a lower bound on the final number of active campaigns could be derived by exploiting the task complexity factors, ψ_i. Mathematically, it can be written as:

$$\sum_{c} X_c \geq \left\lceil \frac{\sum_{i \in ET} \psi_i}{\Psi} \right\rceil \tag{4.22}$$

where ET is the set of essential tasks, *i.e.* all those which lead directly or indirectly to the production of materials, for which the minimum demand over the entire horizon exceeds the initially available amount.

4.5 MILP Tightening

Further enhancements of the MILP model can be achieved by exploiting the minimum campaign duration constraints and also by introducing strong valid inequalities in order to restrict the feasible region of the LP relaxation of the MILP. Consequently, the resulting formulation will present a smaller integrality gap, thus improving the efficiency of the solution method.

4.5.1 Binary Representation of the Number of Cycles

If a minimum campaign duration, $D^{\min}$, is imposed on the solution, then the number of binary variables needed to represent the number of cycles can be reduced, thus reducing the size of the problem in terms of variables and constraints. Furthermore, the resulting MILP formulation is less ill-conditioned with respect to the L_{ic} variables (*cf.* equations (3.3)) as their coefficients are smaller.

For illustration purposes, consider the case of a campaign involving between 100 and 200 cycles. If the minimum number of cycles is not considered, then we must introduce 8 binary variables with coefficients 1, 2, 4, 16, 32, 64, 128 respectively:

$$n_c = L_{0c} + 2L_{1c} + 4L_{2c} + 8L_{3c} + 16L_{4c} + 32L_{5c} + 64L_{6c} + 128L_{7c}$$

On the other hand, if the minimum number of cycles *is* taken into account, then 7 binary variables are sufficient, expressing n_c as:

$$n_c = 100 + L'_{0c} + 2L'_{1c} + 4L'_{2c} + 8L'_{3c} + 16L'_{4c} + 32L'_{5c} + 64L'_{6c}$$

This clearly reduces the number of binary variables by one. However, the important effect is that the largest coefficient introduced is reduced from 128 to 64, which improves the conditioning of the formulation.

In general, if we introduce a minimum number of cycles per campaign $n_c^{\min} = \left\lfloor \frac{D^{\min}}{T_c} \right\rfloor$, then equations (3.3) can be replaced by:

$$n_c = n_c^{\min} X_c + \sum_{i=0}^{r'_c-1} 2^i L'_{ic} \quad \forall\, c \tag{4.23}$$

where $r'_c = \left\lfloor \log_2 \left(n_c^{\max} - n_c^{\min} \right) \right\rfloor + 1$.

4.5.2 Generation of Valid Inequalities

For problems involving interior point events, the duration of each campaign is bounded by the maximum difference between successive distinct point events:

$$n_c T_c \le \max_{\mu} \left(\tau_1,\ \max_{\mu=2..NDP-1} (\tau_\mu - \tau_{\mu-1}),\ H - \tau_{NDP} \right) \tag{4.24}$$

Combining this with (4.23), we can show that:

$$\sum_{i=0}^{r_c-1} 2^i L_{ic} \le \left\lfloor \frac{\max_{\mu} \left(\tau_1,\ \max_{\mu=2..NDP-1} (\tau_\mu - \tau_{\mu-1}),\ H - \tau_{NDP} \right)}{T_c} \right\rfloor - n_c^{\min} \quad \forall\, c \tag{4.25}$$

for which we can generate a set of valid inequalities on L_{ic} (see, *e.g.* Williams (1990), Nemhauser and Wolsey (1988)).

More valid inequalities can be generated when interior point deliveries with lower bounds other than zero are involved in the problem. More specifically, minimum total amounts of material that must be processed by a task $i*$ up to event μ, $\hat{R}^{\mu}_{i*}$, can be determined by solving the following LP:

$$\min \ \hat{R}^{\mu}_{i*}$$

s.t.

$$0 \leq BS^{I}_{s1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{is} \hat{R}^{\mu}_{i} - \sum_{i \in T_s} \rho_{is} \hat{R}^{\mu}_{i} + \sum_{\mu'=1}^{\mu} R^{\max}_{s\mu'} - \sum_{\mu'=1}^{\mu} D^{\min}_{s\mu'} \leq C_s \quad \forall \ s$$

where the above constraint is simply an aggregated material balance on each state s.

Now, if an event μ takes place at the end of campaign c, then the duration of all campaigns up to c must be at least such that the minimum processing amounts as defined above, $\hat{R}^{\mu}_{i*}$, are achieved. Mathematically, the above can be expressed as:

$$\sum_{j \in K_i} \phi^{\max}_{ij} V_j \frac{\sum_{c'=1}^{c} n_{c'} T_{c'}}{p_i} \geq \hat{R}^{\mu}_{i*} \Lambda_{\mu c} \quad \forall \ \mu, c, i \tag{4.26a}$$

We note that the left hand side of the above constraint involves the maximum total capacity of all equipment suitable for task i multiplied by the maximum number of batches of task i that may take place up to the end of campaign c.

From (4.26a) and (4.23), we obtain:

$$\sum_{j \in K_i} \phi^{\max}_{ij} V_j \frac{\sum_{c'=1}^{c} \left(T_{c'} \sum_{k=0}^{r_{c'}-1} 2^k L_{kc'} \right)}{p_i} \geq \hat{R}^{\mu}_{i*} \Lambda_{\mu c} - D^{\min} \frac{\sum_{j \in K_i} V^{\max}_{ij}}{p_i} \sum_{c'=1}^{c} X_{c'}$$

$$\forall \ \mu, c, i \tag{4.26b}$$

All-integer cover constraints may be obtained from (4.26b) using standard procedures (see Appendix B).

Finally, valid inequalities can be derived for the Y^c_i variables from the operability and safety constraints (3.7):

$$\sum_i \psi_i Y^c_i \leq \Psi X_c \quad \forall \ c$$

4.6 Example 1

This is a small problem involving mixed storage policies, multipurpose equipment, sharing of intermediates and material recycles.

Consider a multipurpose batch plant manufacturing three products P1, P2 and P3 according to the following recipe (see figure 4.6):

T1	:	Heat *Feed1* for 2 hours producing *S1*.
T2	:	Blend *S1* immediately after its production with a small amount of additive *Add* for 1 hour to form the unstable intermediate *S2*.
T3	:	Let *S2* react for 1 hour to form the stable intermediate *Int1*.
T4	:	React *Int1* for 2 hours to form intermediate *S3*.
T5	:	Distil *S3* for 2 hours to manufacture *P1*.
T6	:	Separate *S3* for 2 hours to obtain 98% of product *P2* and 2% of *Waste*.
T7	:	Mix 95% of *Feed2* and 5% of *S4* for 4 hours to form intermediate *S5*.
T8	:	Immediately react *S5* for 2 hours to form 90% of the stable intermediate *Int2* and 10% of *S4*, which is recycled to task *T7*.
T9	:	Blend 50% of *Int1* and 50% of *Int2* for 2 hours to form the unstable intermediate *S6*.
T10	:	React *S6* for 3 hour to manufacture product *P3*.

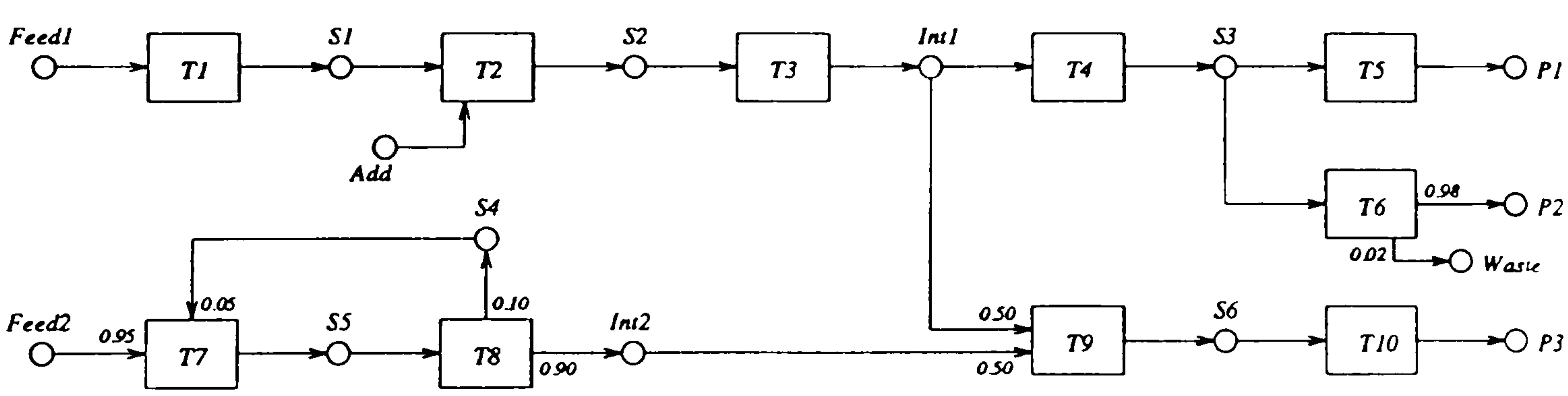


Figure 4.6: STN Representation for Example 1

Unlimited dedicated storage capacity is available for all feeds (*Feed1* and *Feed2*), products (*P1*, *P2* and *P3*) and stable intermediates (*Int1* and *Int2*). Limited amounts of storage capacity are available for *S3* (15 tonnes) and *S4* (40 tonnes). The rest of the states are considered to be unstable, *i.e.* as soon as a batch of material is produced by the upstream task(s), it must immediately be consumed by downstream task(s) (zero-wait operation). The state information is summarized in table 4.1.

State	Initial Amount	Storage Capacity	Unit Value
Feed1	Unlimited	∞	0.0
Feed2	Unlimited	∞	0.0
S1	-	0.0	0.0
S2	-	0.0	0.0
S3	-	15.0 *te*	0.0
S4	-	40.0 *te*	0.0
S5	-	0.0	0.0
S6	-	0.0	0.0
Int1	-	∞	0.0
Int2	-	∞	0.0
P1	-	∞	1.0
P2	-	∞	1.0
P3	-	∞	1.0
Waste	-	∞	0.0

Table 4.1: Details of Material States for Example 1

The available processing equipment is multipurpose in nature as shown in table 4.2.

Unit	Size (tonnes)	Suitable Tasks
Unit1	5.0	*T1, T4*
Unit2	8.0	*T2*
Unit3	6.0	*T3*
Unit4	8.0	*T5, T6*
Unit5	3.0	*T7, T9*
Unit6	4.0	*T8, T10*

Table 4.2: Details of Processing Equipment for Example 1

A minimum batchsize of 25% of their nominal equipment capacity is imposed on all tasks. Irrespective of the processing order, when switching the production between tasks *T1* and *T4*, *T7* and *T9*, and *T8* and *T10* on units 1, 5 and 6 respectively, cleaning of 3 hour duration is needed. No cleaning is required for unit 4.

The objective is to derive a production plan over a horizon of three months (2160 hours), generating a periodic operating schedule within each campaign repeated at a constant cycle time of 12 hours whilst maximising the profit of the plant. Minimum production requirements of 1000 tonnes for product *P1*, and 800 tonnes for products *P2* and *P3* are imposed at the end of the planning horizon.

The maximum potential number of active campaigns is equal to two. Furthermore, the duration of any campaign must be greater than 720 hours (30 days).

One basic characteristic of the algorithm is the exploitation of task coupling (see section 4.1) in order to facilitate the determination of the optimal campaign structure based mainly on the storage available between successive processing tasks. Considering, for instance, state *S2* (for which no storage capacity is available), constraints (4.1a) imply that tasks *T1* and *T2* should co-exist within the same campaign.

Similar couplings can be deduced for states with non-zero but small dedicated storage capacity. For instance, consider state *S3* for which a 15-tonne storage tank is available. Now a possible campaign could involve task $T4$ producing $S3$ but neither of the tasks consuming it (*i.e.* $T5$ and $T6$). However, using the analysis in section 4.1.2, we can show that the minimum nett accumulation of $S3$ per cycle is $\Delta^{L}_{S3,c} = 1.25$ *te*, corresponding to the minimum batchsize in unit 1 $(= 0.25 \times 5)$. As any single campaign must involve at least 60 cycles (=720/12), the minimum nett accumulation of $S3$ per campaign is 75 *te* (1.25×60). Since this exceeds the available storage capacity for $S3$ (15 *te*), we can deduce that this campaign structure is actually infeasible. We can therefore introduce the cut:

$$Y^{c}_{T4} - \left(Y^{c}_{T5} + Y^{c}_{T6} \right) \leq 0 \quad \forall\, c$$

The active parts of the STN for each campaign in the optimal solution are shown in figure 4.7, while the inventory profiles for *Int1* and *Int2* are illustrated in figure 4.8. The total production is 3343.7 tonnes comprising 1583.7, 800.0 and 960.0 tonnes of products *P1*, *P2*, and *P3* respectively. Finally, the durations of the two campaigns are 1152 and 960 hours respectively.

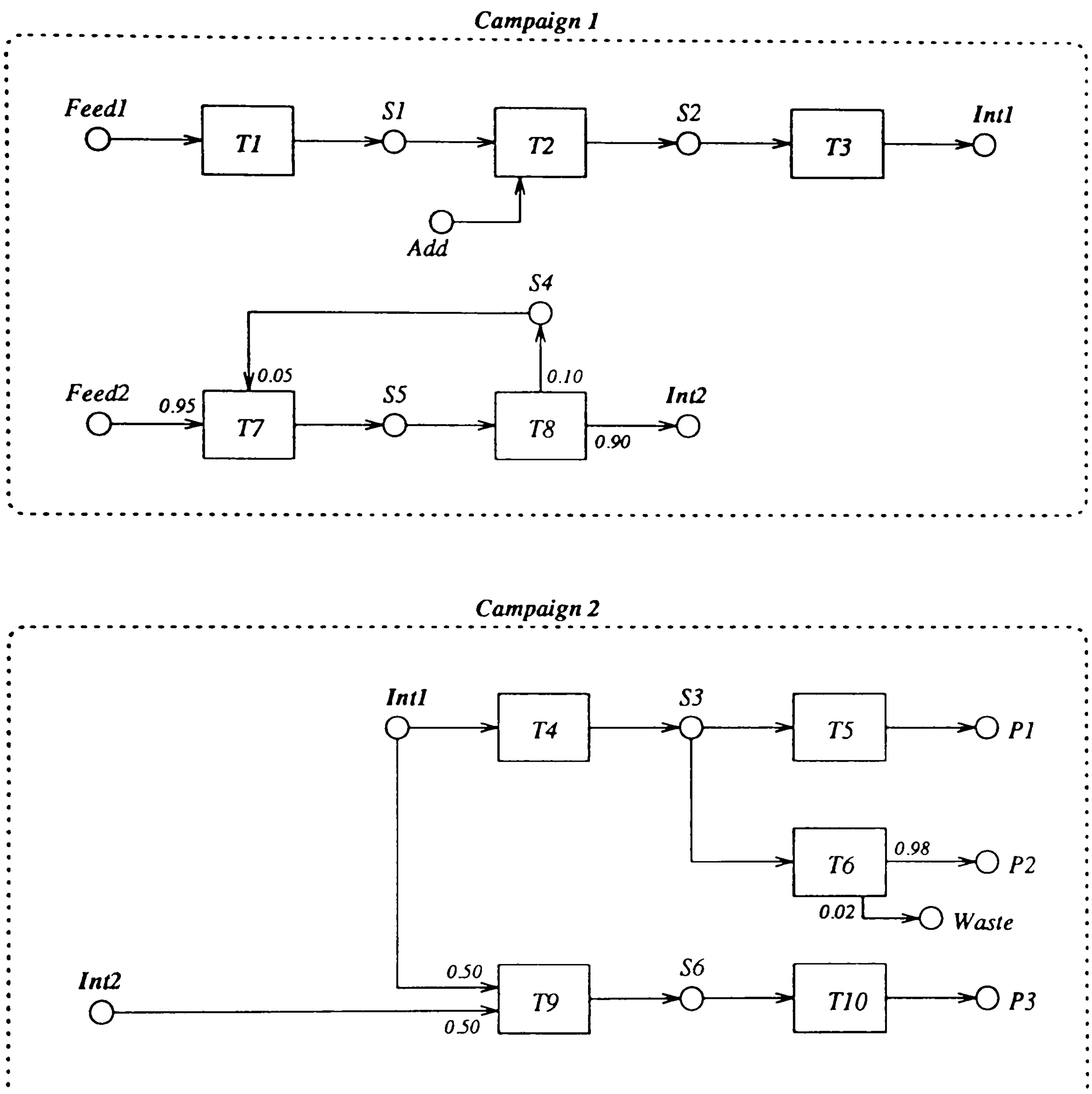


Figure 4.7: Optimal Campaign Structures for Example 1

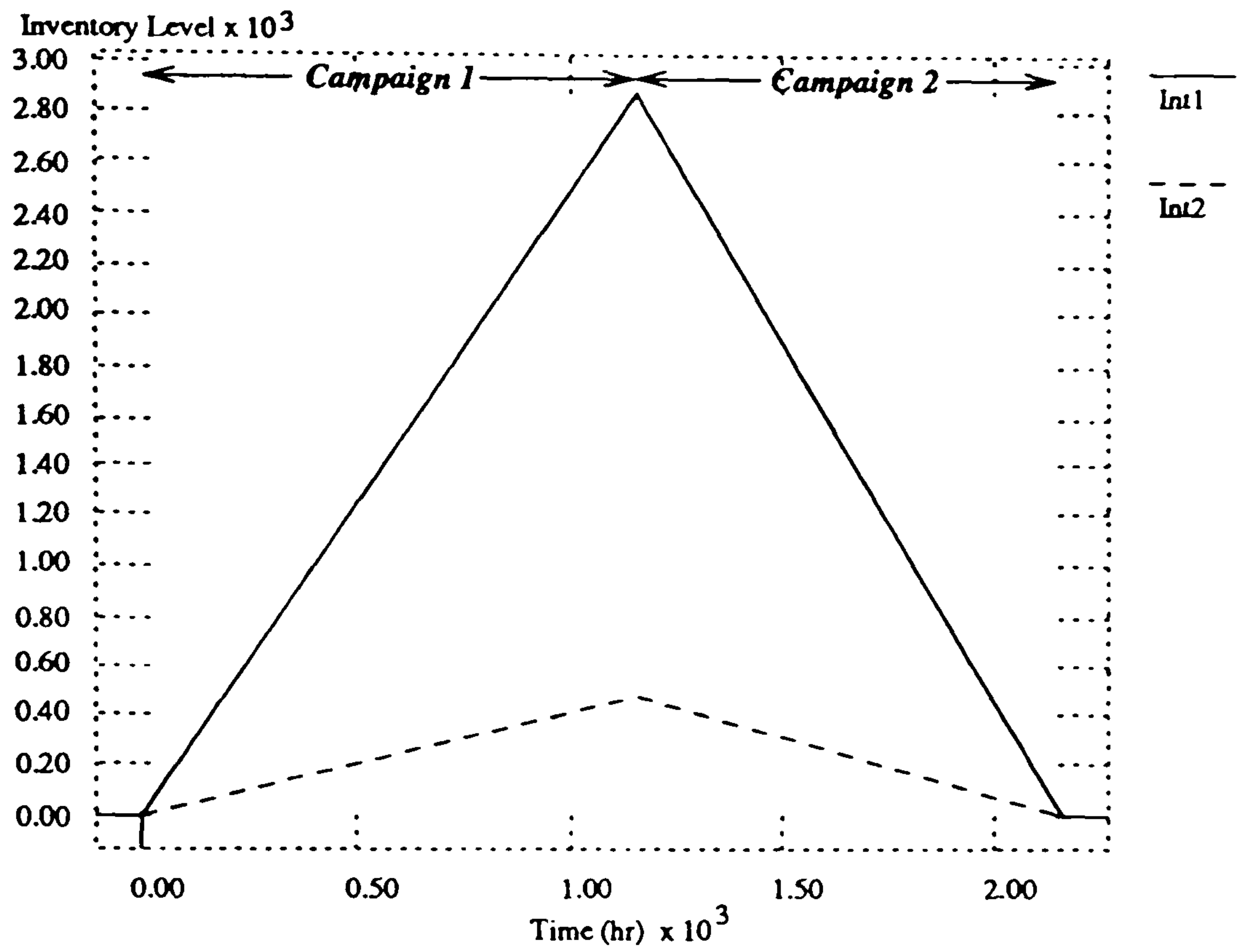


Figure 4.8: Inventory Profiles for Int1 and Int2 for Example 1

Note that during the first campaign, appropriate amounts of *Int1* and *Int2* are built up in order to be consumed by the downstream processes during the second campaign.

The optimal Gantt charts for single cycles of campaigns 1 and 2 are illustrated in figures 4.9 and 4.10 respectively. The number above each horizontal line denotes the processing task, whilst that below is the corresponding batchsize.

Equipment is generally used for a single task over each campaign except for *Unit4* in the second campaign which is used for both *T5* and *T6* without incurring any cleaning requirements. Furthermore, the first campaign is longer than the second mainly due to *Unit1* which constitutes the capacity bottleneck of the plant; it is particularly critical for the production of the coupling intermediate *Int1*.

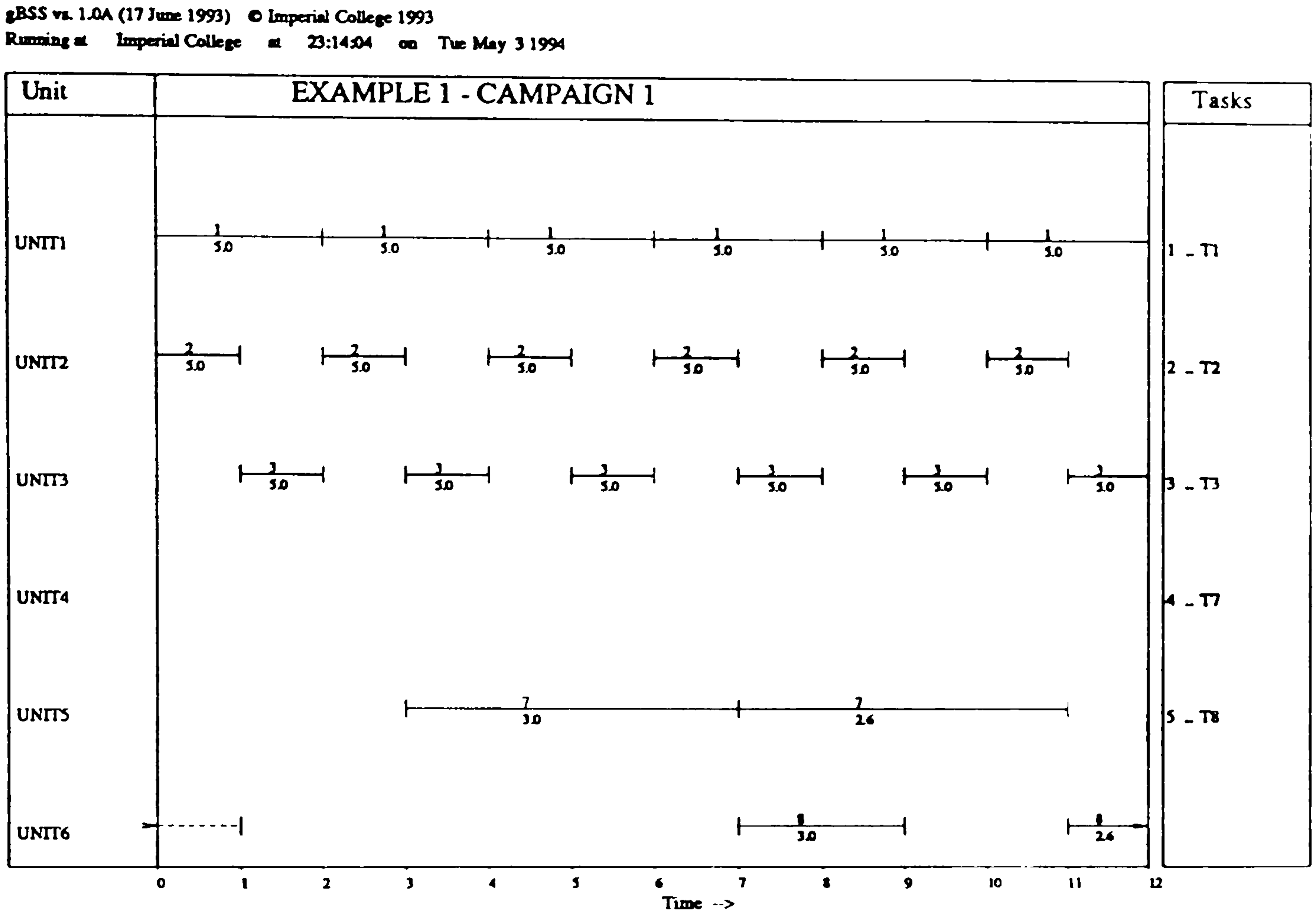


Figure 4.9: Gantt Chart for Campaign 1 for Example 1

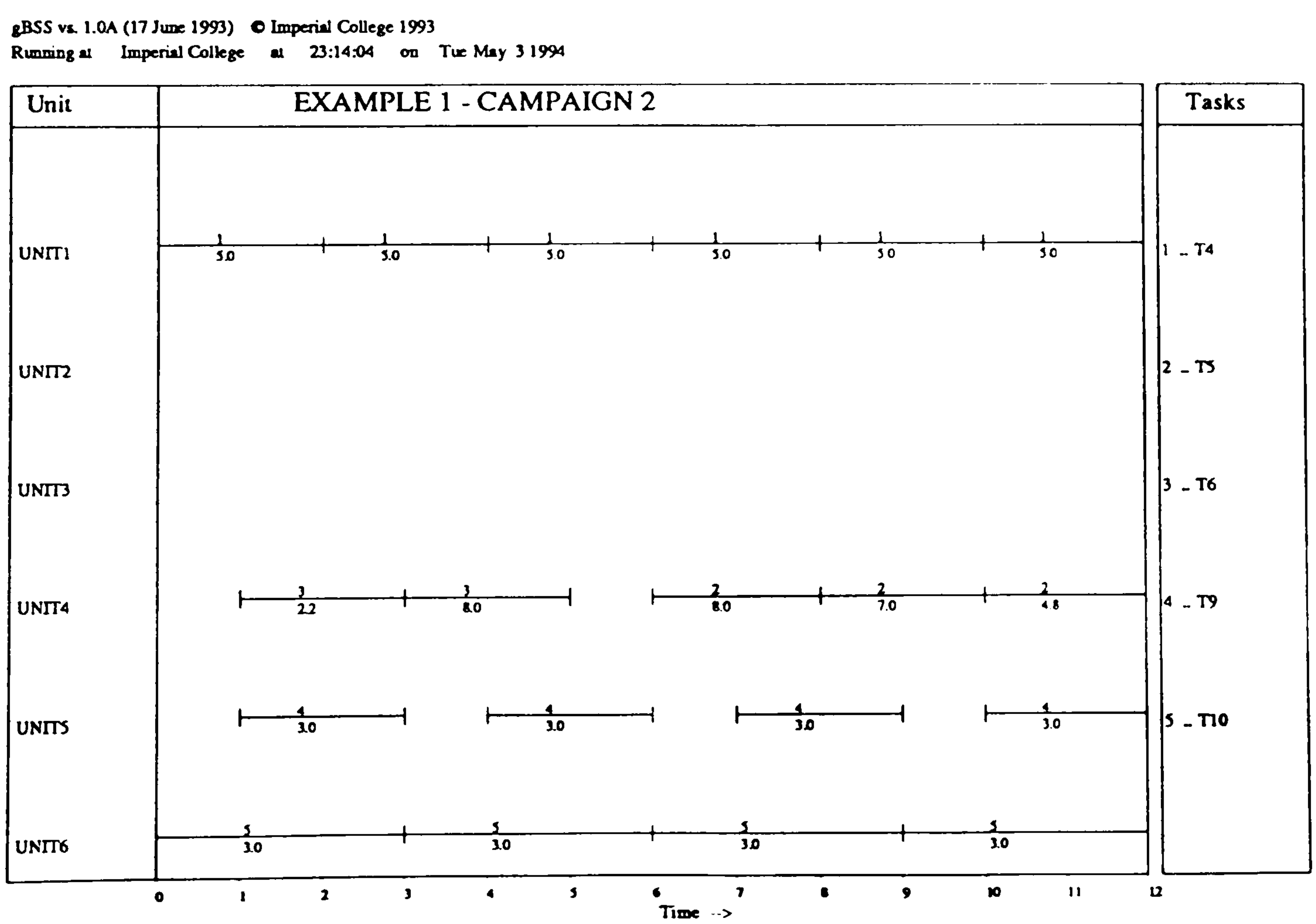


Figure 4.10: Gantt Chart for Campaign 2 for Example 1

Finally, the detailed storage utilisation profile for *S3* over each cycle in the second campaign is illustrated in figure 4.11.

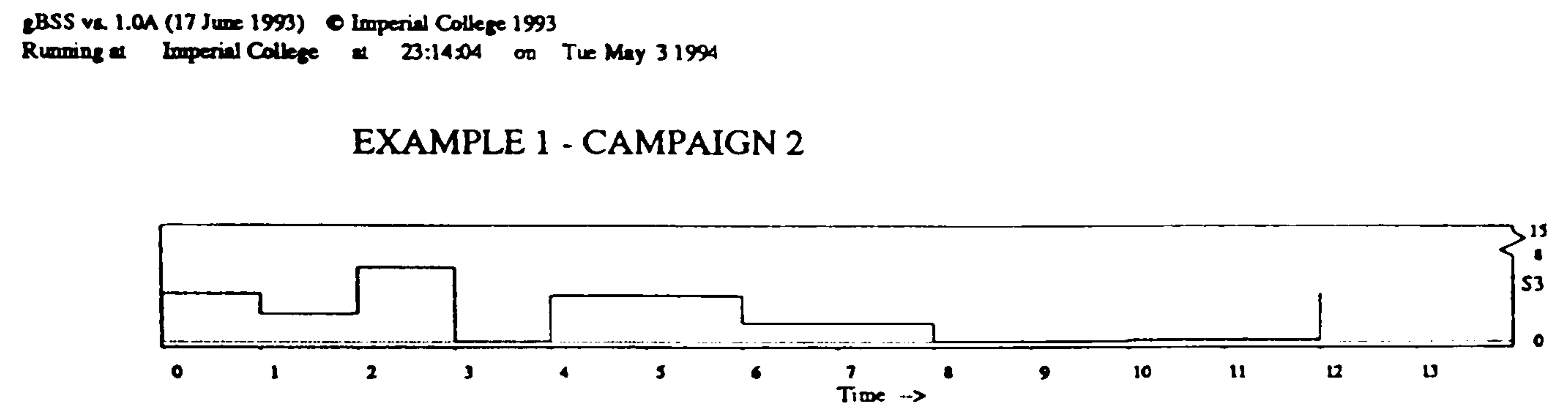


Figure 4.11: Storage Utilisation Profiles for S3 for Campaign 2 for Example 1

This example has been solved to within a relative margin of optimality of 5%, and involves 863 variables (171 binary), and 2048 linear constraints, with a discretisation interval of 1 hour. It is worth mentioning that without applying variable elimination (see section 4.1) the resulting model would involve 959 variables (267 binary). The optimal solution was found after 19774 LPs in 19251 CPUs on a SUN SPARCStation 10/41, using SCICONIC (1990) for the solution of the MILP model.

4.7 Conclusions

This chapter considered various ways of improving the efficiency of the solution of the MILP formulation derived in chapter 3. These centered around the *a priori* elimination of variables, and the introduction of additional redundant constraints aiming to tighten the MILP relaxation.

The generality of the algorithm was demonstrated by a simple example. However, the computational cost of solving the problem as a single MILP remains a major cause of concern even after all the improvements achieved by the techniques introduced in this chapter. We are therefore led to consider, in the next chapter, a mathematical decomposition approach.

Chapter 5

A Mathematical Decomposition Approach to the Campaign Planning/Scheduling Problem

In chapter 3, a single-level mathematical formulation has been presented describing the multiple campaign planning problem, while in chapter 4 general enhancements to the mathematical formulation were described. The resulting MILP model is very large in most of the cases, rendering the problem difficult to solve, as seen in section 4.6.

The basic reasons for the intractability of the single-level formulation are the potentially large number of active campaigns, combined with the time discretisation of the operating cycle within each campaign which results in large numbers of variables and constraints. The complexity of the processing recipes and the multipurpose utilisation of all the resources also contribute to this problem.

Furthermore, the integrality gap of the MILP formulation can be quite substantial. This is primarily caused by the linearisation of the bilinear terms $L_{ic}\Delta S_{sc}$ (see section 3.2.5) using the technique proposed by Glover (1975). In particular, it is difficult to derive tight bounds on ΔS_{sc}, and this, together with the fact that the coefficient 2^i of $L_{ic}\Delta S_{sc}$ in (3.11) (or (3.13a, b)) can result in a large integrality gap in the formulation.

Instead of solving the whole problem simultaneously, we can alternatively adopt a decomposition approach to the solution of the original problem. In general, decomposition approaches often prove to be efficient and rigorous ways of tackling such large models. Of course, selecting the "proper" decomposition of the original problem is an important step for achieving an efficient solution.

The proposed decomposition approach comprises a *master problem* and a *subproblem*. In the master problem, the number of active campaigns, the constituent processing tasks of each campaign and the alignment of possible interior events with campaign boundaries are determined. In the subproblem, the exact duration of, and a detailed operating schedule for each campaign are determined.

The formulation of the subproblem is essentially the same as that presented in chapter 4, with *fixed* values of the X_c, Y_i^c and $\Lambda_{\mu c}$ variables as determined by the master problem. Note, however, that the resulting model may be significantly smaller in size: the W_{ijt}^c variables and the relevant constraints are not generated indiscriminately for every task and every campaign but only for those for which the master problem has set the corresponding Y_i^c

variables to one, thereby including task i in campaign c.

The master problem represents a relaxation of the formulation of chapter 3.

In general, the effectiveness of decomposition approaches depends strongly on the tightness of this relaxation. The tighter this relaxation, the fewer iterations between master problem and subproblem will be required to obtain the optimal solution to the original problem. This tightness may be achieved by involving as much aggregate scheduling information in the master problem as possible. On the other hand, there is a trade-off between the extent of the scheduling description incorporated, and the size of the resulting problem and the computational time required for its solution.

In the rest of this chapter, we consider appropriate formulations first for the master problem and then for the subproblem. We then present the overall decomposition scheme, and demonstrate its effectiveness by applying it to the illustrative example described in section 4.6, and then to two larger ones.

5.1 Master Problem Formulation

5.1.1 Aggregate Campaign Scheduling Formulation for Master Problem

We consider the use of *aggregate* models in order to incorporate scheduling information in the master problem. First, we introduce the following aggregate variables, and indicate their relation to those of the detailed formulation of section 3.3.

$\overline{N}^c_{ij}$ = total number of batches of task i in unit j during a cycle of campaign c $(\overline{N}^c_{ij} \equiv \sum_{t=1}^{T_c} W^c_{ijt})$.

$\overline{B}^c_{ij}$ = total batchsize of task i in unit j during a cycle of campaign c $(\overline{B}^c_{ij} \equiv \sum_{t=1}^{T_c} B^c_{ijt})$.

$\overline{U}^c_u$ = total amount of utility u required during a cycle of campaign c $(\overline{U}^c_u \equiv \sum_{t=1}^{T_c} U^c_{ut})$.

CT^c_j = total cleaning time for unit j in a cycle of campaign c.

Next, the main campaign scheduling constraints included in the master problem are described.

5.1.1.1 Aggregate Equipment Allocation Constraints

The total occupation time of unit j in each cycle of campaign c performing tasks $i \in I_j$ cannot exceed the cycle time T_c minus the required cleaning time for unit j:

$$\sum_{i \in I_j} p_i \bar{N}_{ij}^c \leq T_c X_c - CT_j^c \quad \forall\, c,\, j \tag{5.1}$$

Also, if unit j does not perform task i in campaign c (*i.e.* $Y_{ij}^c = 0$), then the corresponding number of batches ($\bar{N}_{ij}^c$) should be forced to zero. On the other hand, if $Y_i^c = 1$, then unit j must perform *at least one* batch of task i:

$$Y_{ij}^c \leq \bar{N}_{ij}^c \leq \left\lfloor \frac{T_c}{p_i} \right\rfloor Y_{ij}^c \quad \forall\, c,\, j,\, i \in I_j \tag{5.2}$$

5.1.1.2 Aggregate Batchsize Constraints

The total batchsize of task i in unit j during a cycle of campaign c, $\bar{B}_{ij}^c$, is constrained by:

$$\phi_{ij}^{\min} V_j \bar{N}_{ij}^c \leq \bar{B}_{ij}^c \leq \phi_{ij}^{\max} V_j \bar{N}_{ij}^c \quad \forall\, c,\, j,\, i \in I_j \tag{5.3}$$

5.1.1.3 Aggregate Material Balances over a Cycle

An aggregate material balance may be obtained by summing up all the material balance constraints (3.29) over index t:

$$S_{s,T_c+1}^c = S_{s0}^c + \sum_{i \in \bar{T}_s} \sum_{j \in K_i} \bar{\rho}_{si} \bar{B}_{ij}^c - \sum_{i \in T_s} \sum_{j \in K_i} \rho_{is} \bar{B}_{ij}^c \quad \forall\, c,\, s \tag{5.4}$$

5.1.1.4 Aggregate Utility Utilisation Constraints

Also summing up all the utility constraints (3.31) over index t yields:

$$\bar{U}_u^c = \sum_i \sum_{j \in K_i} \left(\bar{N}_{ij}^c \sum_{\theta=0}^{p_i-1} \alpha_{ui\theta} + \bar{B}_{ij}^c \sum_{\theta=0}^{p_i-1} \beta_{ui\theta} \right) \quad \forall\, c,\, u \tag{5.5}$$

In some cases, tighter allocation constraints may be formulated by exploiting special features of the scheduling problem. One such case is that of serial processing networks (*trains*) of tasks operating under a zero-wait mode. Assuming that the equipment items suitable for performing each task in the train are dedicated to it and operate in an out-of-phase mode, we can modify (5.1) to:

$$\bar{N}_{ij}^{c} \max_{i' \in train} \left(\frac{p_{i'}}{card(K_{i'})} \right) \leq T_c X_c - CT_j^c \quad \forall\, c, (i, j) \qquad (5.6)$$

Note that the number of batches for every task in the train is determined by the longest processing task, taking into account the available equipment items for this task.

5.1.1.5 Equipment Cleaning Time

The cleaning time required for every unit j in campaign c is a consequence of the tasks being performed by that unit and the corresponding inter-task cleaning times. The exact length of this time can be ascertained only by examining a detailed operating schedule. However, it is possible to determine a lower bound on it by considering the set of tasks for which the unit is to be used during a given campaign.

In particular, we make the realistic assumption that the cleaning time required between successive batches of the *same* task does not exceed the cleaning time between successive batches of any two *different* tasks. Then the minimum cleaning time is achieved by carrying out all batches of one task successively before switching the unit over to a different task.

Under such conditions, the minimum cleaning time can be determined simply by considering the optimal ordering of the tasks. Consider for instance, a unit j which is suitable for three different tasks with the following cleaning time matrix

		To Task		
		1	2	3
	1	-	2hr	1hr
From Task	2	1hr	-	2hr
	3	1hr	2hr	-

By inspection, the optimal cyclic ordering of the three tasks is 1-3-2-1-3-2... incurring a total cleaning time of 1+2+1=4 hr per cycle.

Of course, the cleaning time per cycle for a given campaign will depend on which of the tasks for which the unit is suitable are actually active over the campaign. Thus, for the above example, if only tasks 1 and 2 are active over a given campaign, then the minimum cleaning time is only 2+1=3 hr per cycle. And if only task 1 is active, then no cleaning time is necessary. We therefore have to enumerate all possible combinations, and determine the optimal cyclic task ordering and the corresponding cleaning time for each:

Combination	Y^c_{1j}	Y^c_{2j}	Y^c_{3j}	Optimal Cyclic Task Ordering	CT^{min}_j (hr)	U_{kj}	D_{kj}
k=1	1	1	1	1-3-2-1-3-2...	4	{1,2,3}	∅
k=2	1	1	0	1-2-1-2...	3	{1,2}	{3}
k=3	1	0	1	1-3-1-3...	2	{1,3}	{2}
k=4	1	0	0	1-1...	0	{1}	{2,3}
k=5	0	1	1	2-3-2-3...	4	{2,3}	{1}
k=6	0	1	0	2-2...	0	{2}	{1,3}
k=7	0	0	1	3-3...	0	{3}	{1,2}
k=8	0	0	0	-	0	∅	{1,2,3}

In general, each combination k is characterised by the set of tasks for which Y^c_{ij} are 1 and those which are 0. We denote these sets by U_{kj} and D_{kj} respectively (see last two columns of the above table). For each combination, we can determine the minimum total change-over, denoted by τ_{kj}. The latter calculation is a relatively small travelling salesman problem which can be solved as a MILP model or by explicit enumeration.

We now introduce the following constraints into the master problem formulation:

$$CT^c_j \geq \tau_{kj}\left[\sum_{i \in U_{kj}} Y^c_{ij} - \sum_{i \in D_{kj}} Y^c_{ij} - card(U_{kj}) + 1\right] + \sum_{i \in I_j} (\bar{N}^c_{ij} - 1)\bar{\tau}_{ij} \quad \forall\, c, j, k \qquad (5.7)$$

where $\bar{\tau}_{ij}$ is the cleaning time required between consecutive batches of the same task carried out on unit j. Often this is zero.

The first term on the right hand side of (5.7) corresponds to cleaning taking place between batches of *different* tasks when switching the unit utilisation from one task to another, while the second term accounts for cleaning incurred between batches of the *same* task. It should be noted that the first term is positive for exactly one combination k, corresponding to the current values of Y^c_{ij} variables.

5.1.2 Campaign Planning Constraints for Master Problem

The campaign planning problem constraints in the master problem are essentially the same as those presented in section 3.2 in the context of the single-level formulation. The main difference is in the way the number of cycles, n_c in each campaign is characterised.

The binary encoding representations (3.3) or (4.23) are certainly the most economical in terms of the number of binary variables variables required. On the other hand, the introduction of large coefficients 2^i may have a detrimental effect on the efficiency of solving the MILP.

The alternative would be to use a representation of the form:

$$n_c = \sum_{i=0}^{n_c^{\max}} i\bar{L}_{ic} \quad \forall\, c \tag{5.8a}$$

or

$$n_c = n_c^{\min} X_c + \sum_{i=0}^{n_c^{\max}-n_c^{\min}} i\bar{L}'_{ic} \quad \forall\, c \tag{5.8b}$$

where the binary variables $\bar{L}_{ic}$ (or $\bar{L}'_{ic}$) form a special ordered set:

$$\sum_{i=0}^{n_c^{\max}} \bar{L}_{ic} = 1 \quad \forall\, c \tag{5.9a}$$

or

$$\sum_{i=0}^{n_c^{\max}-n_c^{\min}} \bar{L}'_{ic} = 1 \quad \forall\, c \tag{5.9b}$$

However, this may result in the introduction of too many binary variables if $n_c^{\max} \gg n_c^{\min}$.

For the purposes of the master problem, we note that it is not essential to characterise the number of cycles in each campaign exactly. Instead we attempt to determine n_c only within a range of a given width, ΔN. In particular, we partition the maximum number of cycles in a campaign, $n_c^{\max}$, into a number of intervals, each (except possibly the last one) involving the same number of cycles, ΔN:

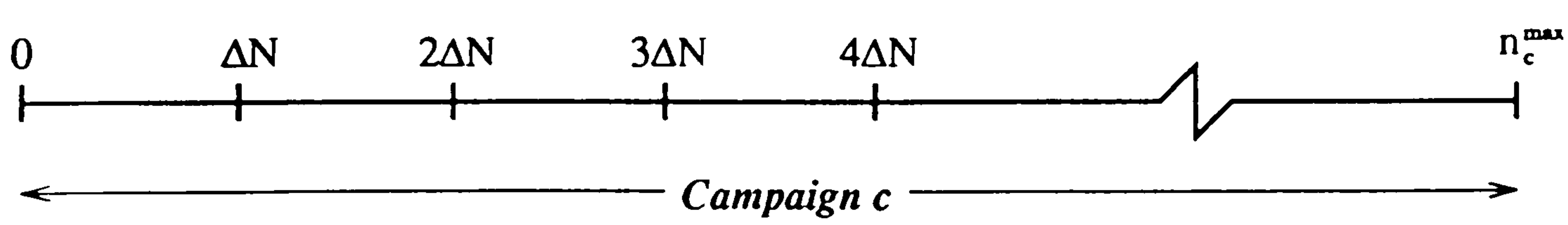


Figure 5.1: Time Description

We now introduce binary variables, $\bar{L}_{ic}$, such that:

$$\bar{L}_{ic} \quad = \quad 1 \text{ if } (i-1)\Delta N \le n_c < i\Delta N;\ 0 \text{ otherwise.}$$

for $i = 1..\left\lceil \dfrac{n_c^{\max}}{\Delta N} \right\rceil$.

These variables clearly form a special ordered set:

$$\sum_i \bar{L}_{ic} = 1 \quad \forall\, c \tag{5.10}$$

and we can write:

$$\Delta N \sum_i (i-1)\bar{L}_{ic} \le n_c < \Delta N \sum_i i L_{ic} \quad \forall\, c \tag{5.11}$$

Thus, we can bound the duration of each campaign, $D_c \equiv n_c T_c$, by:

$$T_c \Delta N \sum_i (i-1)\bar{L}_{ic} \le D_c < T_c \Delta N \sum_i i\bar{L}_{ic} \quad \forall\, c \tag{5.12}$$

We now consider the bilinear products $n_c \Delta S_{sc}$ appearing in the material balance constraints (3.10). The quantity ΔS_{sc} can be either positive or negative depending on whether there is a nett production or consumption of state s in campaign c. However, it can always be represented as the difference of two non-negative quantities, ΔS^+_{sc} and ΔS^-_{sc}:

$$\Delta S_{sc} = \Delta S^+_{sc} - \Delta S^-_{sc} \quad \forall\, c, s \tag{5.13}$$

Here ΔS^+_{sc} and ΔS^-_{sc} can be interpreted as the amounts of material in state s respectively produced and consumed over a cycle in campaign c.

We therefore have:

$$n_c \Delta S_{sc} = n_c \Delta S^+_{sc} - n_c \Delta S^-_{sc} \quad \forall\, c, s \tag{5.14}$$

Now from (5.11) and since $\Delta S^+_{sc} \ge 0$ we can derive:

$$\Delta N \sum_i (i-1)\bar{L}_{ic}\Delta S^+_{sc} \le n_c \Delta S^+_{sc} < \Delta N \sum_i i\bar{L}_{ic}\Delta S^+_{sc} \quad \forall\, c, s \tag{5.15a}$$

and also from (5.11) and since $\Delta S^-_{sc} \ge 0$ we have:

$$\Delta N \sum_i (i-1)\bar{L}_{ic}\Delta S^-_{sc} \le n_c \Delta S^-_{sc} < \Delta N \sum_i i\bar{L}_{ic}\Delta S^-_{sc} \quad \forall\, c, s \tag{5.15b}$$

Introducing new variables $S^+_{ics} \equiv \bar{L}_{ic}\Delta S^+_{sc}$ and $S^-_{ics} \equiv \bar{L}_{ic}\Delta S^-_{sc}$, and combining (5.14), (5.15a) and (5.15b), we obtain the desired bounds on the quantity $n_c \Delta S_{sc}$:

$$\Delta N \left(\sum_i (i-1) S^+_{ics} - \sum_i i S^-_{ics} \right) < n_c \Delta S_{sc} < \Delta N \left(\sum_i i S^+_{ics} - \sum_i (i-1) S^-_{ics} \right) \quad \forall\, c, s \tag{5.16}$$

All that remains is to define the new variables, S^+_{ics} and S^-_{ics}, through linear constraints:

$$0 \le S^+_{ics} \le M^+_{ics} L_{ic} \quad \forall\, c, s, i \tag{5.17a}$$

and

$$0 \le S^-_{ics} \le M^-_{ics} L_{ic} \quad \forall\, c,\, s,\, i \tag{5.17b}$$

where M^+_{ics} and M^-_{ics} are appropriate upper bounds on ΔS^+_{sc} and ΔS^-_{sc} respectively. Constraints (5.17a, b) ensure that S^+_{ics} and S^-_{ics} are zero if $L_{ic} = 0$.

Finally, the following constraints:

$$\Delta S_{sc} = \sum_i \left(S^+_{ics} - S^-_{ics} \right) \quad \forall\, c,\, s \tag{5.18}$$

derived from (5.13) and (5.10) relate these quantities to ΔS_{sc}.

The linear constraints (5.10), (5.16), (5.17a, b) and (5.18) can be used to eliminate the bilinear product $n_c \Delta S_{sc}$ from the material balances. For instance, (3.10) can be written as:

$$\Delta N \left(\sum_i (i-1) S^+_{ics} - \sum_i i S^-_{ics} \right) < BS^F_{sc} - BS^I_{sc} \quad \forall\, c,\, s \tag{5.19a}$$

and

$$BS^F_{sc} - BS^I_{sc} < \Delta N \left(\sum_i i S^+_{ics} - \sum_i (i-1) S^-_{ics} \right) \quad \forall\, c,\, s \tag{5.19b}$$

5.2 Subproblem Formulation

As mentioned earlier in the introduction of this chapter, the formulation of the subproblem is similar to the single-level mathematical formulation presented in chapter 3 with the X_c, Y^c_i and $\Lambda_{\mu c}$ variables fixed at the values determined by the master problem.

Compared to the single-level formulation of chapter 3, the subproblem formulation is much smaller as every campaign involves only a portion of the original STN and also the number of active campaigns under consideration may be much smaller than the potentially available campaigns (*i.e.* N). Thus the integer W^c_{ijt} variables will be fewer and the computational requirements will be more modest.

As the master problem involves only aggregate information on scheduling aspects, it constitutes a relaxation of the feasible region of the original problem. During the solution of the subproblem, the algorithm may effectively reduce the number of campaigns in the problem by setting the corresponding n_c to zero.

One interesting point concerns the determination of the campaign lengths, or equivalently the number of cycles in each campaign. As seen in section 5.1, the master problem allows campaign durations to vary between ranges. A reasonable approach for the *subproblem* would then appear to be to determine the exact number of cycles from within the range determined by the master problem. This would have the advantage that, since ΔN would normally be relatively small, a special ordered set representation could be used for n_c (*cf.* equation (5.8a, b)) instead of the less well-conditioned binary encoding one (equation (3.3)).

However, in practice it was found to be better to ignore the range determined by the master problem for the purposes of solving the subproblem. This was due to the fact that very often the master problem determines n_c to lie in (slightly) wrong ranges, and enforcing these in the subproblem simply results in more master-subproblem iterations in the decomposition scheme (see section 5.3).

5.3 Decomposition Scheme

As the master problem constitutes a relaxation of the original problem, its solution is a valid upper bound, Φ^U, on the value of the optimal objective function, Φ. On the other hand, the subproblem provides a lower bound Φ^L on Φ as its feasible region is restricted with respect to the original one due to the fixed values of variables determined by the master problem.

At every iteration of the decomposition, previous assignments of binary variables determined by previous master problems are excluded through appropriate cut constraints on these binary variables at every iteration, k. More specifically, at the Kth iteration we introduce the following cuts:

$$\sum_{(c,i,\mu)\in U^{(k)}} \left(X_c^{(k)} + Y_i^{c(k)} + \Lambda_{\mu c}^{(k)} \right) - \sum_{(c,i,\mu)\in L^{(k)}} \left(X_c^{(k)} + Y_i^{c(k)} + \Lambda_{\mu c}^{(k)} \right) \le card(U^{(k)}) - 1$$

$$\forall\, k = 1..K-1 \tag{5.20}$$

where $U^{(k)}$, $L^{(k)}$ denote the subsets of the X_c, Y_i^c and $\Lambda_{\mu c}$ variables that were fixed to their upper and lower bounds respectively at the solution of the master problem at iteration k. We note that these cuts involve *only* the binary variables whose values are fixed during the subproblem solution.

When the lower and upper bounds converge to within a prespecified tolerance, the algorithm terminates with a value of the objective function Φ^{IP} and the best feasible solution (BFS).

In summary, the solution algorithm comprises the following steps.

Given a margin of optimality, ε:

1. Set $\Phi^{IP} := -\infty$. $k := 0$.
2. Solve the master problem (including cuts (5.20)) obtaining, if feasible, an upper bound, Φ^U. The branch-and-bound search for this step is initialised using Φ^{IP} as the best objective function value found so far, as master problem solutions with $\Phi^U < \Phi^{IP}$ are not useful (*cf.* step 4).
3. If the master problem is infeasible, the algorithm terminates. If $\Phi^{IP} := -\infty$ then the original problem is infeasible as well, otherwise the BFS is the optimal solution with an objective function value of Φ^{IP}.
4. If $(\Phi^U - \Phi^{IP})/\Phi^{IP} \le \varepsilon$ then the algorithm terminates with a ε margin of optimality.
5. Solve the subproblem with the variables determined by the master problem fixed, determining campaign durations and detailed operating schedule for every campaign, and a valid objective function lower bound Φ^L. The branch-and-bound search for this step is initialised using Φ^{IP} as the best objective function value found so far, as subproblem solutions with $\Phi^L < \Phi^{IP}$ are not useful (*cf.* step 6).
6. If $\Phi^L > \Phi^{IP}$ keep the current solution as BFS and set $\Phi^{IP} := \Phi^L$.
7. If $(\Phi^U - \Phi^{IP})/\Phi^{IP} \le \varepsilon$ then the algorithm terminates with a ε margin of optimality.
8. Set $k := k + 1$. Go to step 2.

Overall, the decomposition scheme adopted is shown diagrammatically in figure 5.2.

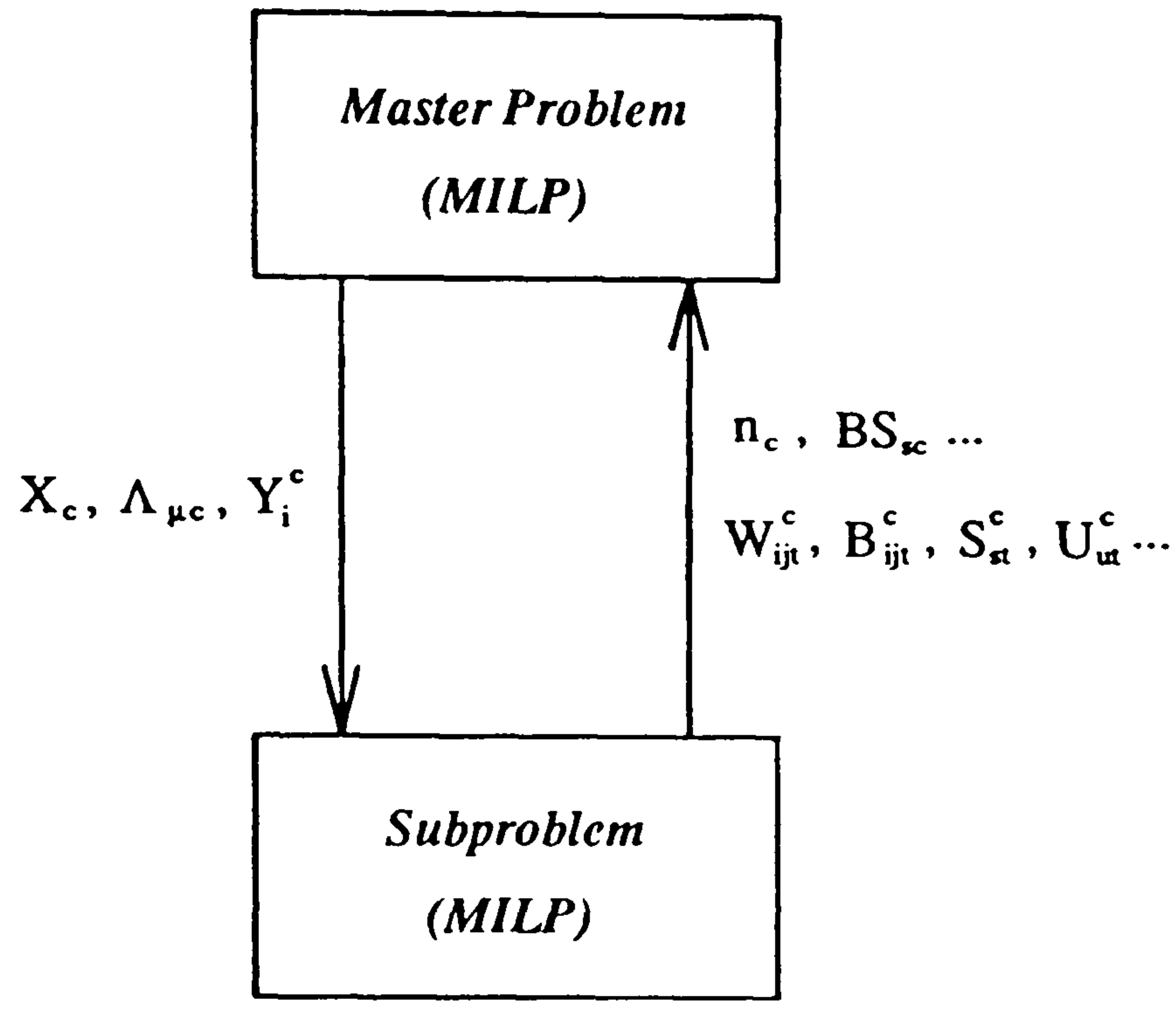


Figure 5.2: Decomposition Scheme

5.4 Example 1 Revisited

Here, we re-consider the example presented in section 4.6. The two potential campaigns can accommodate up to 180 cycles (2160 hours) and 120 cycles (1440 hours) respectively, or 8 and 6 intervals each of 20 cycles (*i.e.* $\Delta N = 20$).

The solution of this problem using the decomposition approach presented in the previous section required two master problems and one subproblem to be solved. The first master problem iteration results in the following solution:

Campaign	Range of n_c	Active Tasks
1	[60 80]	T1, T2, T3, T7, T8
2	[80 100]	T4, T5, T6, T9, T10

Table 5.1: First Master Problem Solution for Example 1

The corresponding subproblem determines the exact number of cycles to be $n_1 = 95$ and $n_2 = 84$ (*i.e* $D_1 = 1140$ *hr* and $D_2 = 1008$ *hr*). It can be seen that n_1 is actually outside the range [60, 80] determined by the master problem. This occurs because of the relaxation utilised in the master problem which tends to under-estimate the consumption of feedstocks and over-estimate the production of products.

The campaign structures obtained from the master problems and subproblems respectively are given in table 5.2.

	MP	SP
Campaign 1	T1, T2, T3, T7, T8	T1, T2, T3, T7, T8
Campaign 2	T4, T5, T6, T9, T10	T4, T5, T6, T9, T10

MP: Master Problem, and SP: Subproblem

Table 5.2: Campaign Structures for Example 1

Table 5.3 gives the relevant computational details for the decomposition scheme, along with the results presented in section 4.6 (single-level solution) for comparison. Essentially the first iteration determines the optimal solution to the problem as the second master iteration gives an infeasible solution, thus terminating the procedure (step 3 of the algorithm).

	DECOMPOSITION APPROACH			SINGLE-LEVEL SOLUTION
	Iteration 1		Iteration 2	
	MP	SP	MP	
NIV	37	93	37	171
NOV	325	517	325	863
NOC	633	981	634	2048
LPs	75	98	109	19774
CPU	30.7	21.3	23.6	19291.0
OBJ	3899.5	3318.5	Infeasible	3325.4

key:
- NIV : Number of Integer Variables
- NOV : Number of Variables
- NOC : Number of Constraints
- LPs : LP Calls for MILP Solution (SCICONIC)
- CPU : Cpu Time (sec) on SUN SPARCStation 10/41
- OBJ : Objective Function Value (relative cost units — *rcu*)

Table 5.3: Computational Details for Example 1

Table 5.3 illustrates the efficiency of the decomposition approach presented in this chapter. It can be seen that the optimal campaign structure is obtained from the first master iteration. Furthermore, both the total number of LPs and the computational time required have been reduced by approximately two orders of magnitude compared to the single-level solution approach presented in chapter 4.

5.5 Example 2

This example emphasises trade-offs between inter-task and inter-campaign change-overs, also taking account of inventory costs.

We consider a multiproduct plant manufacturing three products, *P1*, *P2* and *P3* from three feedstocks, *Feed1*, *Feed2* and *Feed3* over a planning horizon of four months (2880 hours). Minimum amounts of 400 tonnes (*te*) of *P1* and 500 tonnes of *P2* and *P3* are required by the end of the 4-month period. Continuous demands of 0.1 and 0.2 *te/hr* for *P1* and *P2* respectively, must be satisfied throughout the planning horizon.

Unlimited amounts of all three feedstocks as well as 200 tonnes of *P2* are available at the start of the horizon. A cycle time of 12 hours is used. A minimum campaign duration of 720 hours is imposed on the plant operation.

The inventory cost calculations are based on an interest rate value of 10% and a price of 1 *rcu/te* for all products.

The following processing steps are involved in the process, corresponding to the STN shown in figure 5.3.

T10	:	React feedstock *Feed1* to produce *S10* after 4 hours.
T11	:	Filter *S10* for 2 hours to manufacture *S11*.
T12	:	Dry immediately *S11* for 2 hours to produce final product *P1*.
T20	:	React feedstock *Feed2* to produce *S20* after 3 hours.
T21	:	Filter *S20* for 2 hours to manufacture *S21*.
T22	:	Dry immediately *S21* for 2 hours to produce final product *P2*.
T30	:	React feedstock *Feed3* to produce *S30* after 2 hours.
T31	:	Filter *S30* for 1 hour to manufacture *S31*.
T32	:	Dry immediately *S31* for 2 hours to produce final product *P3*.

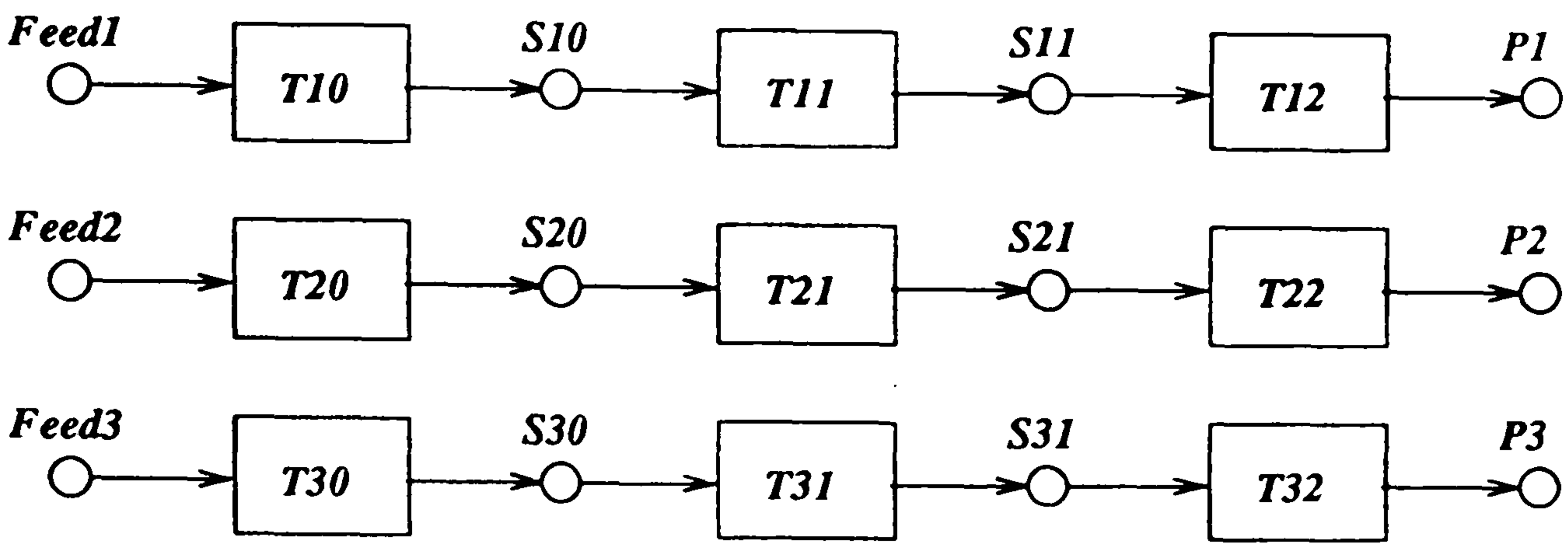


Figure 5.3: STN for Exampe 2

Unlimited storage is available for all feedstocks and final products. Three 10-tonne storage tanks dedicated to *S10*, *S20* and *S30* respectively, are available to buffer the production between reaction and filtering steps.

The processing equipment is shown in table 5.4.

Unit	Size (*te*)	Suitable Tasks
Unit1	5.0	*T10, T20, T30*
Unit2	2.0	*T11, T21, T31*
Unit3	3.0	*T12, T22, T32*

Table 5.4: Details of Processing Equipment for Example 2

A minimum batchsize of 25% of the nominal capacity for all processing equipment items is required.

5.5.1 Case I: No Equipment Cleaning Requirements

This case is modelled as a single-campaign problem and the single-level solution approach, presented in chapters 3 and 4, is applied. The mathematical formulation involves 468 variables (122 binary) which are reduced to 432 variables (86 binary) after eliminating variables and constraints using the techniques of section 4.1. Also, the model involves 745 linear constraints.

The problem was solved via a branch-and-bound technique (SCICONIC) with a 5% margin of optimality. The optimal solution was obtained after examining 154 LPs and a CPU time of 39.4 s on a SUN SPARCStation 10/41.

An optimal objective function value of 2129.3 *rcu* was obtained. The final production amounts of the products after deducting the continuous rate deliveries are 652.4, 572.8 and

932.0 tonnes for *P1*, *P2* and *P3* respectively. An inventory cost of 27.9 *rcu* was incurred.

The optimal operating schedule is shown in figure 5.4.

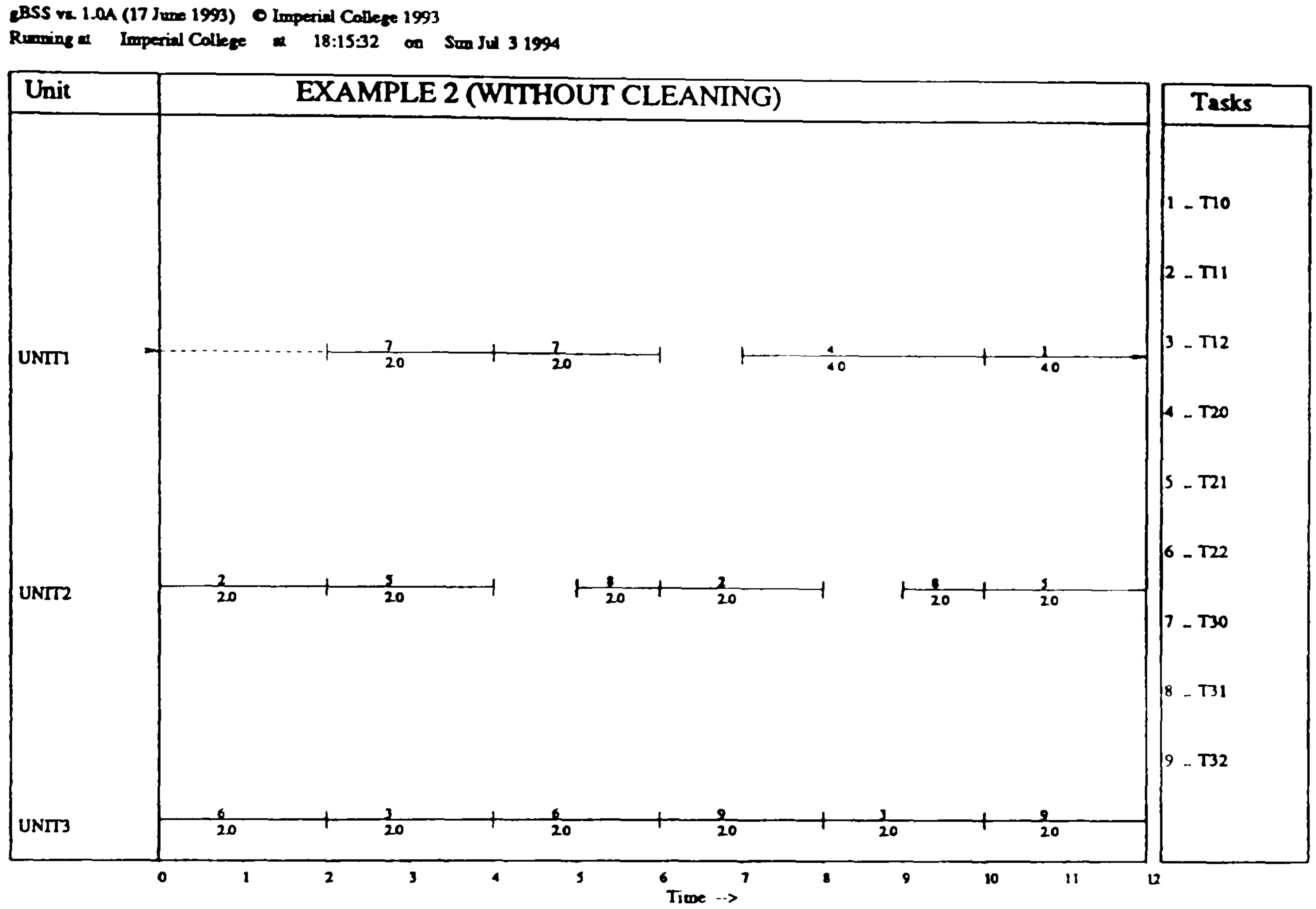


Figure 5.4: Gantt Chart for Example 2 without Equipment Cleaning

5.5.2 Case II: Equipment Cleaning Requirements

Due to the multipurpose nature of the processing equipment, cleaning may be needed when switching the production from one processing task to another. Here, irrespective of the processing order, a cleaning duration of 2 hours is required when switching the production between tasks *T12* and either *T22* or *T32* on *Unit3*, and between *T20* and *T30* on *Unit1*, while 4 hours cleaning is imposed between tasks *T10* and either *T20* or *T30* on *Unit1*.

If we try to solve the above problem using a single campaign as in section 5.5.1, we find that it is infeasible. This is due the fact that a large proportion of the cycle time is taken up by the cleaning, thereby not leaving sufficient time for the required minimum production to be achieved. Next, we consider up to three campaigns with a fixed cycle time of 12 hours. A constant inter-campaign change-over of 240 hours is imposed for plant reconfiguration.

The single-level formulation involves 1447 variables (370 binary) which are reduced to 1339 variables (262 binary) after eliminating variables and constraints using the techniques of section 4.1. The formulation also involves 3379 linear constraints. For its solution, we adopt the decomposition scheme presented in section 5.3.

We use a value of ΔN of 20 cycles for the master problem, and a discretisation interval of 1 hour for the subproblem. All the MILP models are solved via a branch-and-bound procedure (SCICONIC) with a 5% margin of optimality. The problem required 6 master and 5 subproblem iterations for its solution. The computational details are described in table 5.5.

	Iteration 1		Iteration 2		Iteration 3		Iteration 4		Iteration 5		Iteration 6
	MP	SP	MP	SP	MP	SP	MP	SP	MP	SP	MP
NIV	66	95	66	127	66	101	66	127	66	127	66
NOV	507	478	507	650	507	524	507	644	507	650	507
NOC	955	904	956	1235	957	949	958	1215	959	1243	960
LPs	84	80	170	7	198	1	171	1	205	7	371
CPU	50.4	15.3	65.3	14.4	72.1	5.1	67.0	8.2	78.0	13.3	115.9
OBJ	2517.1	1919.6	2228.1	Infeas	2220.6	Infeas	2228.1	Infeas	2228.1	Infeas	Infeas

Table 5.5: Computational Details for Example 2

As can be seen from table 5.5, only the first subproblem is feasible, the rest fail to determine a solution better than that of the first subproblem (1919.6) which is used to fathom branches during the search in step 5 of the algorithm. If the latter is removed, these subproblems become feasible with the same or worse objective function than the first subproblem. For example, the subproblem of the third iteration results in an objective function value of 1702.0 *rcu* which is much smaller than the solution already obtained by the solution of the first subproblem.

The optimal plan involves 2 active campaigns out of 3 potential ones with campaign durations equal to 732 and 1872 hours respectively. The campaign structures determined by the master iterations along with the corresponding ranges of campaign durations are shown in table 5.6.

	CAMPAIGN 1		CAMPAIGN 2		CAMPAIGN 3	
Iteration	Structure	Duration Range (hrs)	Structure	Duration Range (hrs)	Structure	Duration Range (hrs)
1	*T10, T11, T12*	[720-960]	*T20, T21, T22, T30, T31, T32*	[1680-1920]		
2	*T10, T11, T12*	[720-960]	*T20, T21, T22, T30, T31, T32*	[960-1200]	*T20, T21, T22*	[720-960]
3	*T10, T11, T12*	[720-960]	*T20, T21, T22*	[960-1200]	*T30, T31, T32*	[720-960]
4	*T10, T11, T12*	[720-960]	*T20, T21, T22*	[720-960]	*T20, T21, T22, T30, T31, T32*	[960-1200]
5	*T10, T11, T12*	[720-960]	*T20, T21, T22, T30, T31, T32*	[960-1200]	*T30, T31, T32*	[720-960]

Table 5.6: Campaign Structures for Example 2

Figure 5.5 shows the variation of the product inventories during the planning horizon.

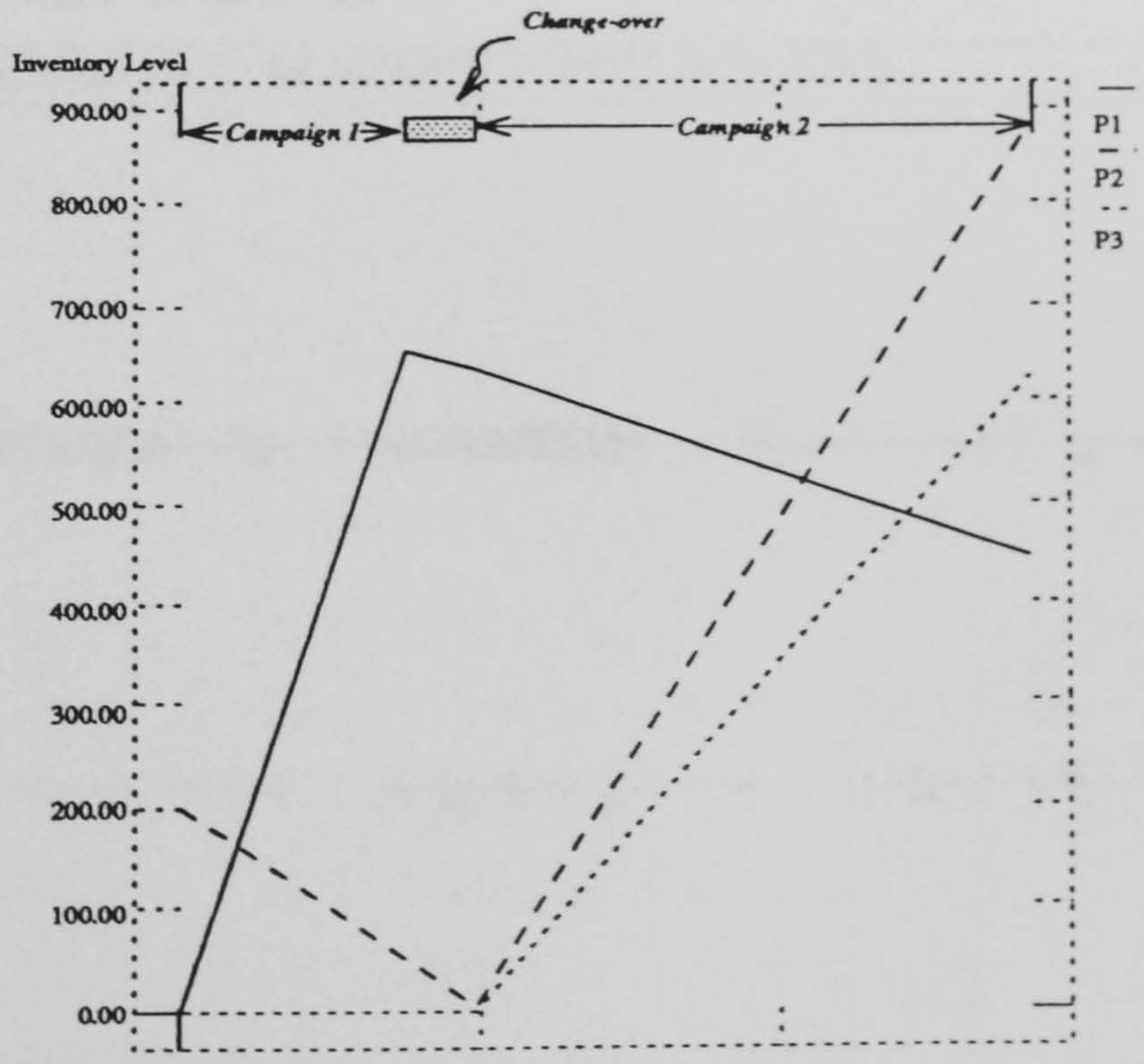


Figure 5.5: Inventory Profiles for P1, P2 and P3 for Example 2

At the end of the 4-month period the amounts of *P1*, *P2* and *P3* manufactured are 447.6, 879.2 and 624.0 tonnes respectively, while the associated inventory cost is 31.2 *rcu*.

The optimal operating schedules (Gantt charts) for a cycle of each of the final two campaigns are illustrated in figures 5.6 and 5.7. The hatched boxes denote cleaning between successive processing tasks.

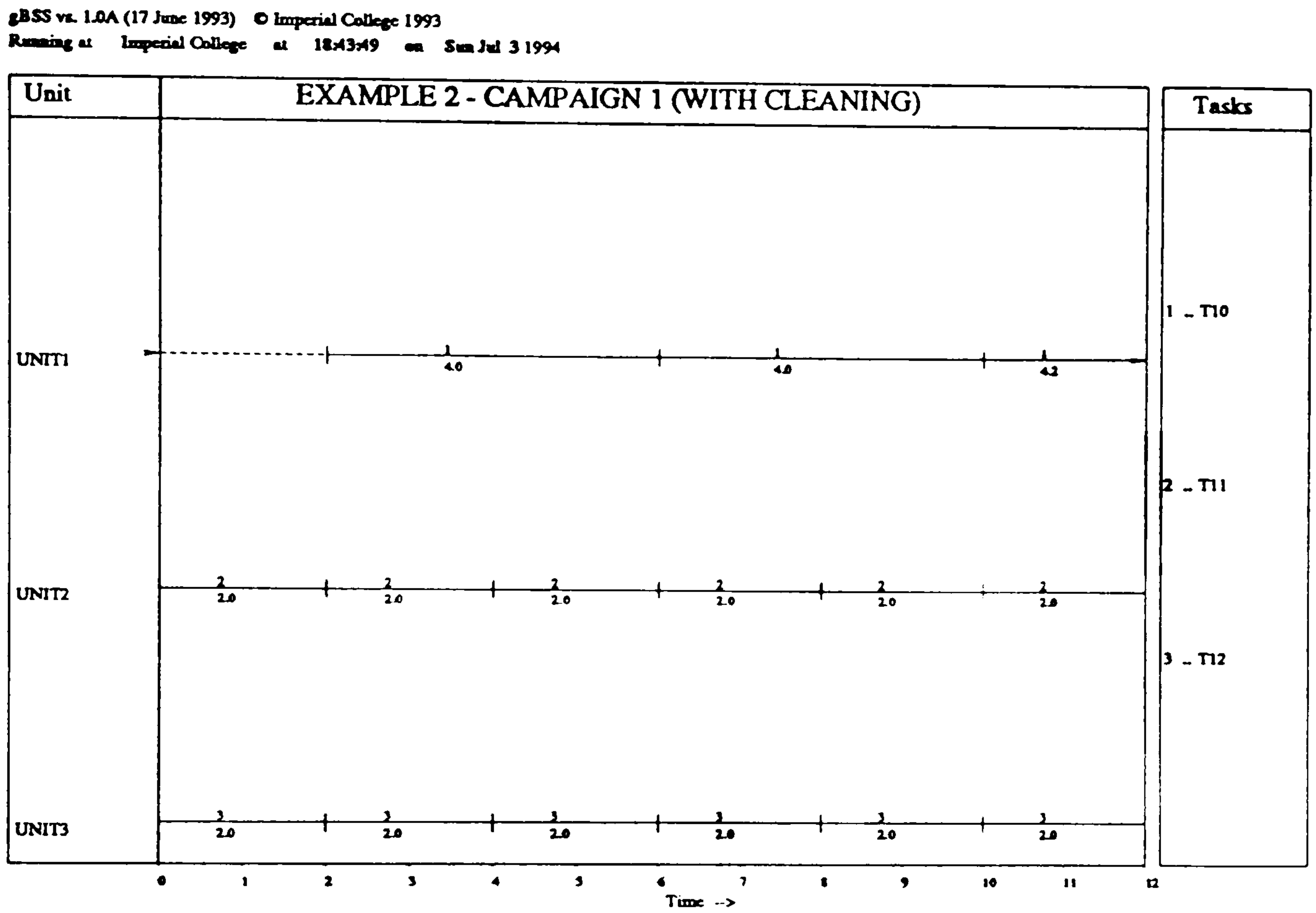


Figure 5.6: Gantt Chart for Campaign 1 for Example 2 with Equipment Cleaning

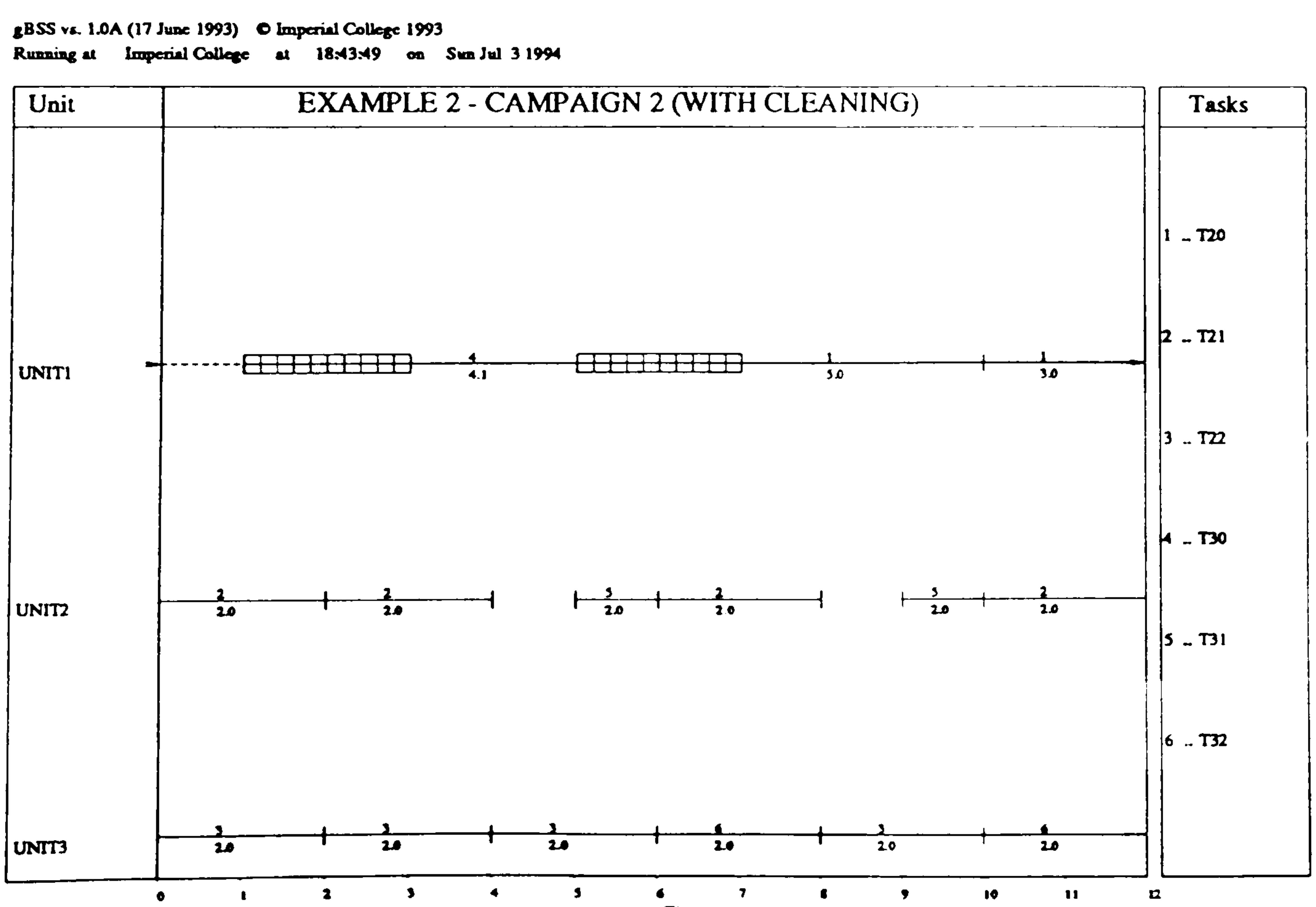


Figure 5.7: Gantt Chart for Campaign 2 for Example 2 with Equipment Cleaning

It can be seen from the results that the first campaign involves the production of *P1* only, while both *P2* and *P3* are produced during the second campaign. This structure avoids most, but not quite all, cleaning requirements. The continuous demand for *P2* during the first campaign is satisfied from the initial stock.

5.6 Example 3

Here, we consider planning the production for a plant manufacturing four products from six feedstocks over a horizon of four months (2880 hours). Minimum amounts of 750 tonnes of each of the final products are required at the end of the 4-month period; however the market demand for all of them is practically unlimited. There are up to four potential campaigns with a given cycle time of 12 hours. Also, a minimum campaign duration of 480 hours is imposed. Fixed inter-campaign change-overs of 240 hours are required when production switches from one campaign to another for plant reconfiguration.

The plant involves 19 processing steps manufacturing four final products as follows (see STN of figure 5.8):

T10 : Blend feedstock *Feed2* to produce *S10* after 3 hours.

T11 : React immediately *S10* for 2 hours to manufacture stable intermediate *Int1*.

T20 : Blend feedstock *Feed1* for 3 hours to produce intermediate *S20*.

T21 : Blend 60% of *Int1* and 40% of *Int2* to form *S21* after 2 hours.

T22 : React for 2 hours 15% of *S20* with 85% of *S21* to obtain *S22*.

T23 : React *S22* for 2 hours to form the final product *P1*.

T30 : React feedstock *Feed3* to obtain intermediate *S30* after 1 hour.

T31 : Blend feedstock *Feed4* for 2 hours to form intermediate *S31*.

T32 : React 80% of *S31* with 20% of *S30* to form two stable intermediates, 50% of *Int2* and 50% of *Int3* after 3 hours.

T40 : Blend intermediate *Int2* to form *S40* after 2 hours.

T41 : React *S40* to produce final product *P2* after of 1 hour.

T50 : Blend intermediate *Int3* to form *S50* after 2 hours.

T51 : React *S50* to produce final product *P3* after of 1 hour.

T60 : React feedstock *Feed6* for 2 hours to produce *S60*.

T61 : React feedstock *Feed5* for 3 hours to form intermediate *S61*.

T62 : Blend 25% of *S60* with 75% of *S61* for 2 hours to manufacture stable intermediate *Int4*.

T70 : Blend 35% of *Int3* with 65% of *Int4* to form *S70* after 2 hours.

T71 : React immediately *S70* for 1 hour to obtain intermediate *S71*.

T72 : React *S71* for 1 hour to produce 95% of final product *P4* and 5% of *S72*.

Unlimited dedicated storage is available for all feedstocks and final products as well as for stable intermediates *Int1*, *Int2*, *Int3* and *Int4*. Furthermore, dedicated limited storage (FIS) of 20 tonnes is available for intermediates *S30*, *S31* and *S61*, and also 10 tonnes for *S20* and *S71*.

The processing equipment is multipurpose in nature as shown in table 5.7.

Unit	Size (*te*)	Suitable Tasks
Unit1	6.0	*T10, T21*
Unit2	5.0	*T32, T41, T51*
Unit3	7.0	*T31, T72*
Unit4	7.0	*T23, 760, T30*
Unit5	8.0	*T40, T50, T20*
Unit6	6.0	*T61, T70*
Unit7	7.0	*T11, T22, T41*
Unit8	8.0	*T62, T71, T51, T72*

Table 5.7: Details of Processing Equipment for Example 3

Several operating constraints are imposed on the utilisation of the equipment items. A minimum batchsize of 25% of the nominal capacity is required for all the processing units. Furthermore, when *switching* the production from one processing task to another, cleaning may be needed. A cleaning duration of 4 hours is imposed on *Unit2* and *Unit3*, 3 hours is imposed on *Unit1*, *Unit5* and *Unit6*, and also 1 hour on *Unit4*. Although *Unit7* and *Unit8* are, in principle, suitable for 3 and 4 different tasks respectively, they are allowed to be used for at most 2 of these in any single campaign in order to facilitate the management and the control of the plant.

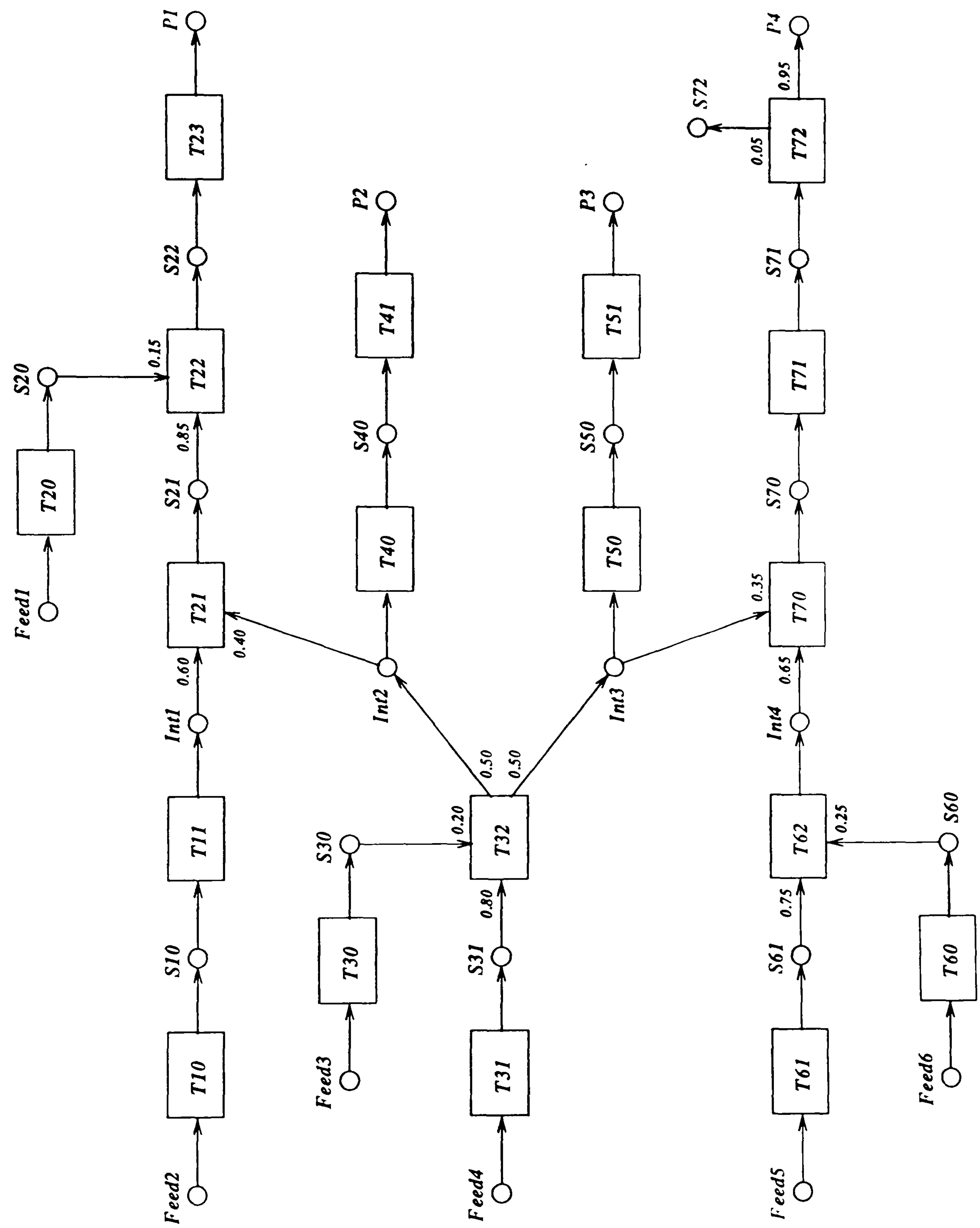


Figure 5.8: STN for Example 3

The single-level formulation involves 4166 variables with 1164 binary variables. These are reduced to 3926 and 924 respectively after elimination of variables using the techniques of section 4.1. The formulation also involves 10125 linear constraints. Due to the large size of the problem, we only attempt to solve it using the decomposition approach.

We used a value of ΔN of 20 cycles for the master problem, and a discretisation interval of 1 hour for the subproblem. All the MILP problems were solved via a branch-and-bound technique (SCICONIC) with a 5% margin of optimality. In total, only 1 master and 1 subproblem iterations were required for the solution of the above problem. The computational details are described in table 5.8.

	MP	SP
NIV	166	432
NOV	1544	2020
NOC	3038	4714
LPs	1157	88
CPU	7170.0	850.2
OBJ	9260.6	9071.9

Table 5.8: Computational Details for Example 3

Note that the algorithm terminates after the first subproblem due to the termination criterion in step 7 (see section 5.3).

In this example, there is a trade-off between inter-campaign change-overs for plant reconfiguration forcing a smaller final number of active campaigns, and inter-task change-overs for cleaning requirements for more active campaigns with fewer constituent processing tasks. The optimal plan comprises 2 active campaigns out of 4 potential ones, with campaign durations of 996 and 1620 hours respectively. The campaign structures and the corresponding campaign durations or ranges obtained during the solution procedure are given in table 5.9.

	MP	SP
CAMPAIGN 1		
Structure	T10, T11, T30, T31, T32, T50, T51, T60, T61, T62	T10, T11, T30, T31, T32, T50, T51, T60, T61, T62
Duration (hrs)	480-720	996
CAMPAIGN 2		
Structure	T20, T21, T22, T23, T30, T31, T32, T40, T41, T70, T71, T72	T20, T21, T22, T23, T30, T31, T32, T40, T41, T70, T71, T72
Duration (hrs)	960-1200	1620
CAMPAIGN 3		
Structure	T20, T21, T22, T23, T30, T31, T32, T50, 51, T70, T71, T72	
Duration (hrs)	240-480	0
CAMPAIGN 4		

Table 5.9: Campaign Structures for Example 3

We can see from the above table that the campaign structure estimated by the master problem is refined by the subproblem by eliminating the third campaign.

The active parts of the original STN involved within each campaign are shown in figure 5.9.

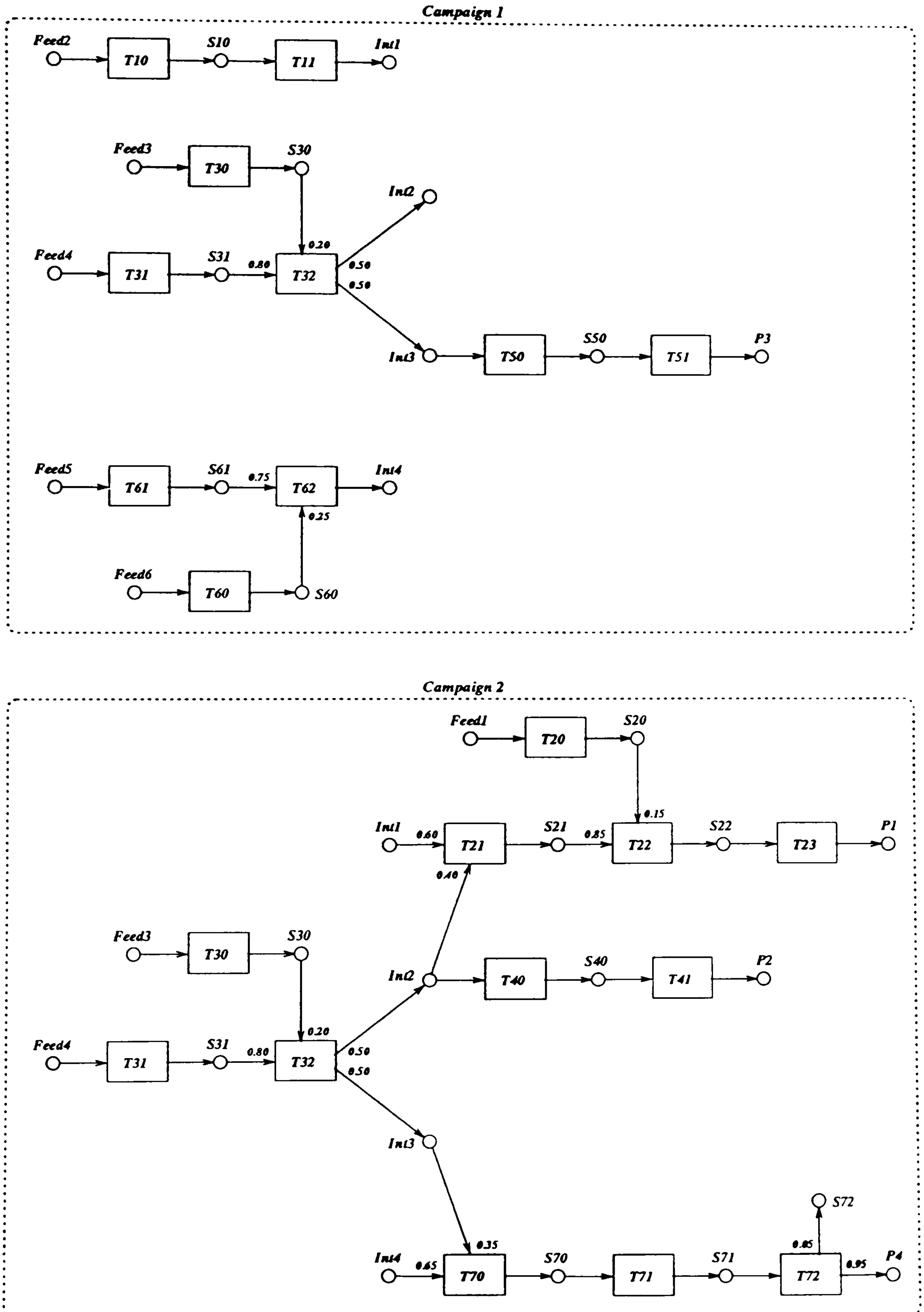


Figure 5.9: Optimal Solution for Example 3

It can be seen from figure 5.9 that product *P3* is manufactured during the first campaign while the rest of the products are produced during the second campaign as illustrated in figure 5.10.

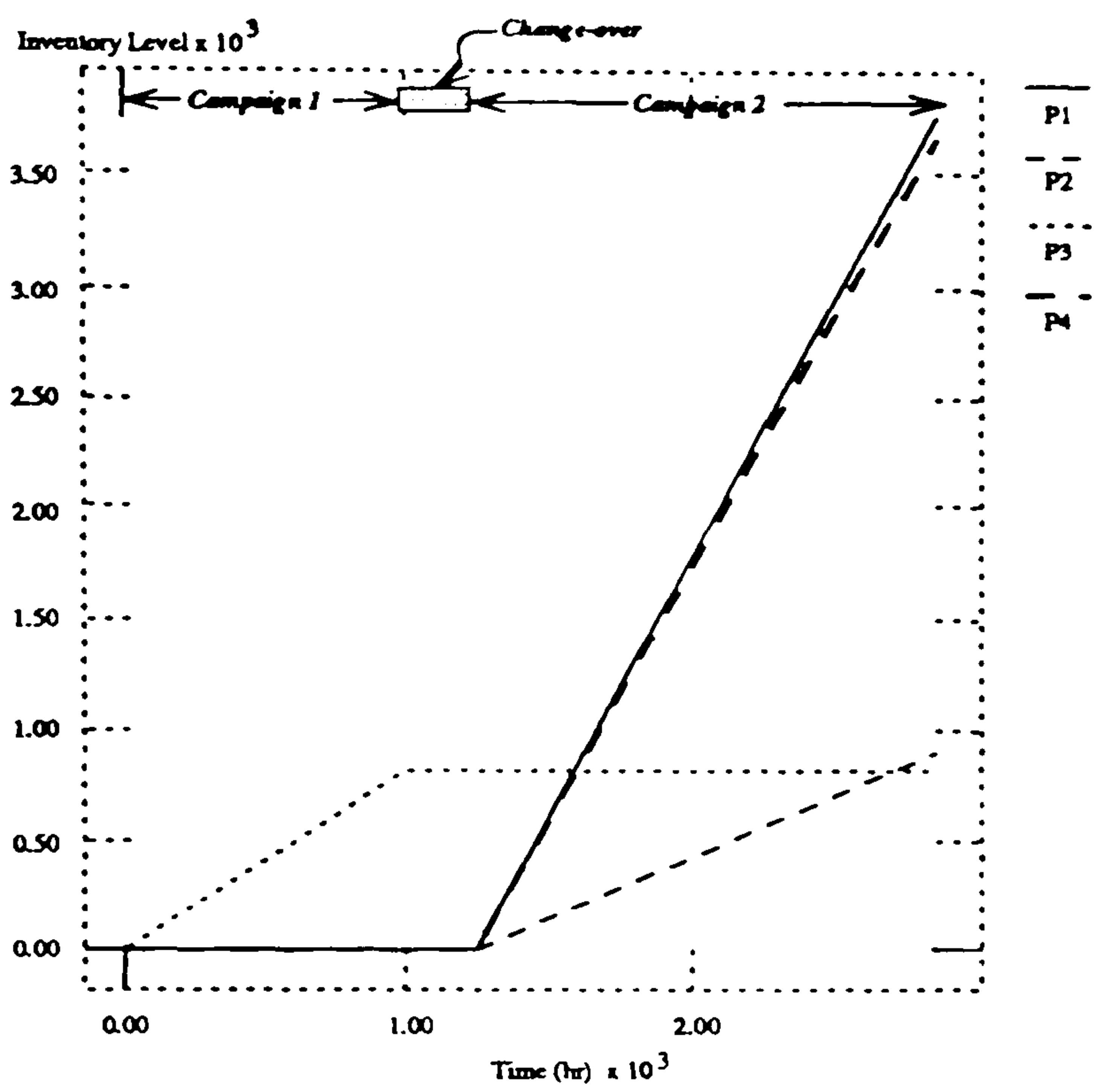


Figure 5.10: Inventory Profiles for P1, P2, P3 and P4 for Example 3

At the end of the planning horizon, the amounts of the final products, *P1*, *P2*, *P3* and *P4*, manufactured are 3780.0, 894.8, 830.0 and 3664.3 tonnes respectively.

Furthermore, during the first campaign intermediates *Int1* and *Int4* are produced, to be consumed by downstream processes during the second campaign. In addition, *Int2* and *Int3* are both produced and consumed in both campaigns. Figure 5.11 shows the inventory profiles over the 4-month planning horizon for four intermediates; *Int1*, *Int2*, *Int3* and *Int4*.

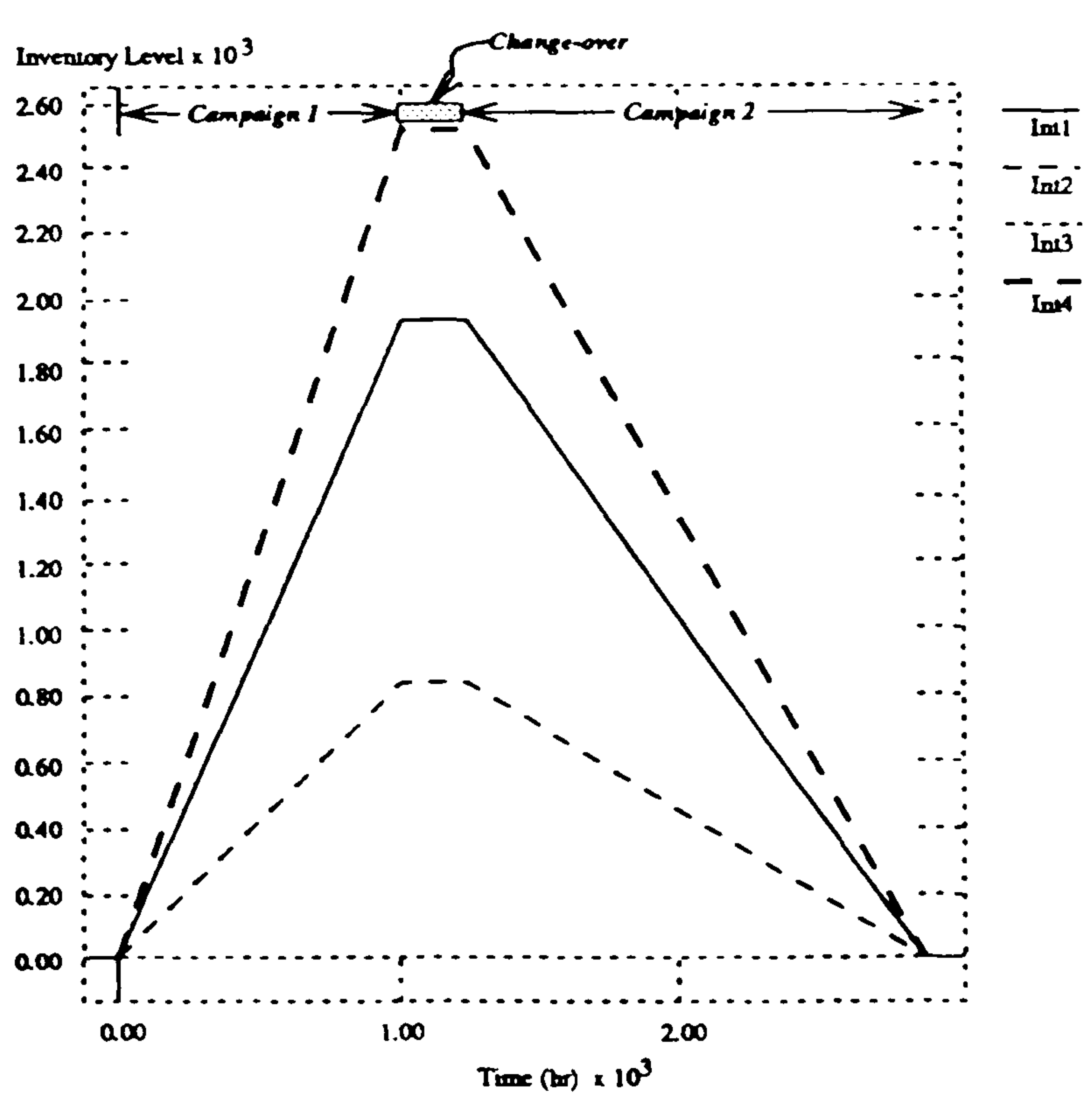


Figure 5.11: Inventory Profiles for Int1, Int2, Int3 and Int4 for Example 3

The optimal operating schedules (Gantt charts) for a cycle of each campaign are illustrated in figures 5.12 and 5.13. The hatched boxes denote cleaning between successive processing tasks.

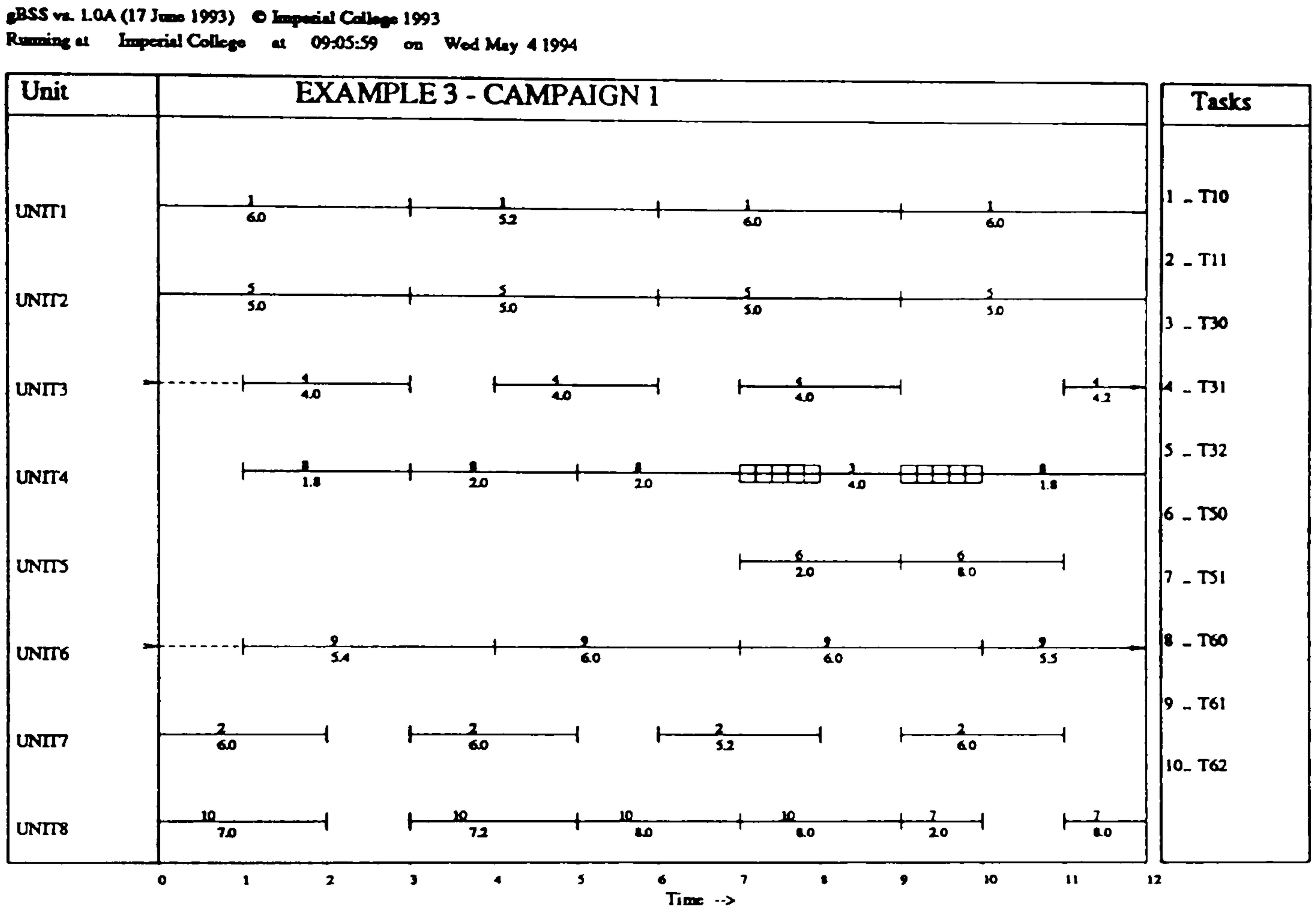


Figure 5.12: Gantt Chart for Campaign 1 for Example 3

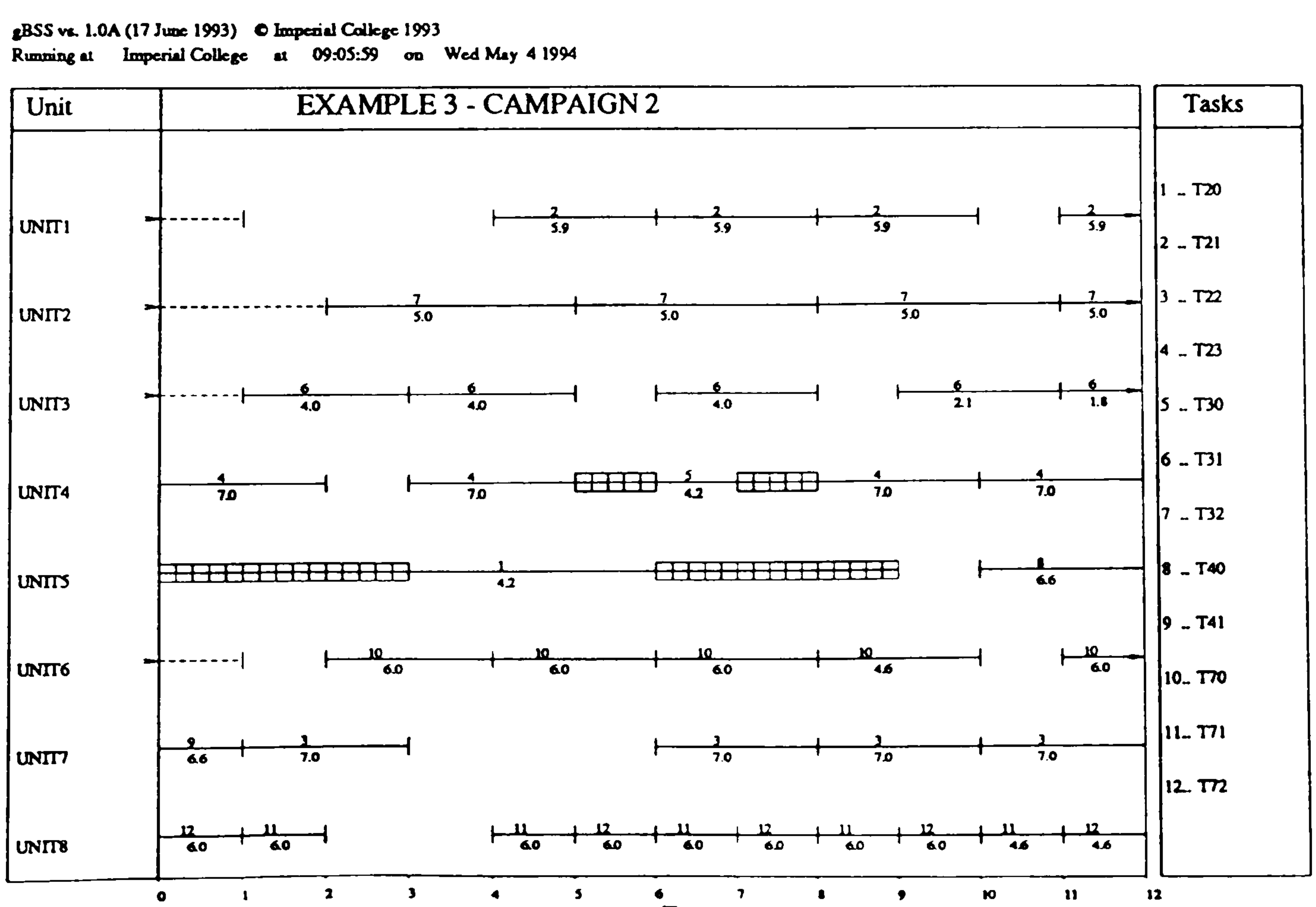


Figure 5.13: Gantt Chart for Campaign 2 for Example 3

The limited amounts of dedicated storage for various states may be used as temporary buffers, thus alleviating production bottlenecks. Such bottlenecks may occur because of limited levels of scarce resources (*e.g.* processing equipment) along with different operating conditions of successive tasks (*e.g.* different processing times, cleaning requirements). For instance, *Unit5* at the second campaign (see figure 5.13) can perform tasks *T20* and *T31* only once due to the large processing time of the former task and also to the large cleaning time required between these tasks. Hence, limited dedicated storage after task *T20* (*i.e.* state *S20*) would enhance the performance of the plant. Figure 5.14 illustrates the detailed storage utilisation profile for *S20* during the second campaign.

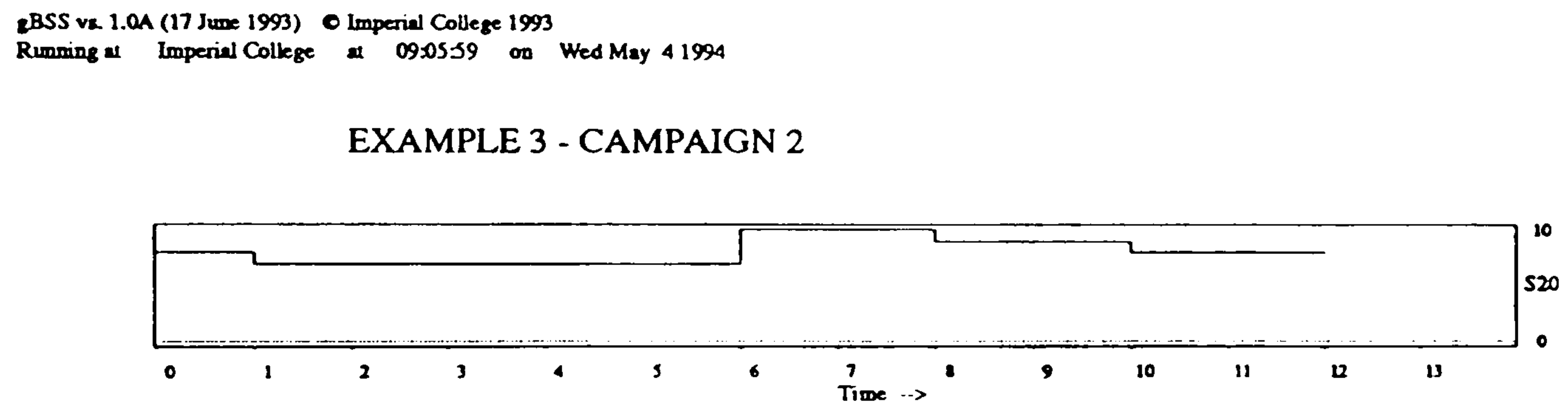


Figure 5.14: Storage Utilisation Profile for S20 for Campaign 2 for Example 3

The optimal value of the objective function is found to be equal to 9071.9 *rcu*, with 9169.1 *rcu* associated with nett profit from the value of the products and 97.2 *rcu* related to the corresponding inventory costs.

5.7 Conclusions

A decomposition approach to the solution of the campaign planning/scheduling problem has been presented. The number of the active campaigns over the whole planning horizon and the constituent processing tasks within each campaign are estimated by the master problem. This solution is subsequently refined by the subproblem. The duration as well as the detailed operating schedule for each campaign are also determined during the solution of the subproblem.

If the original problem is feasible, the solution obtained by the decomposition model corresponds to the global optimum solution to the original planning problem within a margin of optimality ε. In the case of an infeasible original problem, the decomposed model will also be infeasible.

Overall, the decomposition scheme renders large problems tractable within acceptable computing effort. Improvements of up to two orders of magnitude on computational requirements on both LPs and CPU time have been observed.

PART II

ENERGY INTEGRATION IN MULTIPURPOSE BATCH PLANT OPERATION

Chapter 6

Optimal Scheduling of Energy Integrated Multipurpose Plants

The following two chapters address the problem of the optimal exploitation of energy integration in the operation of multipurpose batch/semi-continuous plants. We propose a hierarchical approach with two levels. At the higher level the existence of pairs of energy-integrated batch operations is taken into account within detailed optimal scheduling formulations, thus considering simultaneously energy integration and plant operation. At this level the operating conditions of the energy integrated coupled tasks, *i.e.* the durations, starting time offsets, and total external utility demand profiles are assumed to be known, having been determined during the lower level of the hierarchy.

It must be emphasised that this two-step hierarchical approach does not necessarily guarantee the optimality of the overall plant operation despite the optimality of each of the two steps involved in it.

The rest of this chapter deals with the optimal scheduling of energy integrated plants, considering the complex interaction, between plant scheduling on one hand, and energy integration on the other. The next chapter considers techniques for the optimisation of the operation of energy integrated tasks.

6.1 Literature Review

Recent work (Vaselenak *et al.*, 1986; Lovell-Smith and Baldwin, 1988) has highlighted the need for efficient utilisation of energy in the operation of batch and semi-continuous plants. However, in contrast to the extensive amount of work already published on energy integration in continuous plants, relatively little has been reported in the literature on this aspect of the operation and design of flexible multipurpose plants.

Vaselenak *et al.* (1986) consider the problem of determining the best pairing between batch vessels containing hot fluids that require cooling, and cold fluids that require heating over a given time period. They examine two cases: in the first, the fluids return to the original holding vessels after heat exchange, while in the second, the fluids remain entirely within the holding vessels while heating and cooling media accomplish the exchange. They present a heuristic procedure to determine best pairings when final temperatures are not limiting, and a mixed integer linear programming (MILP) formulation for the case when they are. A given schedule of operations is implicit in the problem description.

Ivanov *et al.* (1992), and Peneva *et al.* (1992) consider the problem of designing a minimum total cost heat exchanger network for given pairwise matches of batch vessels. They examine the cases where heat exchange takes place directly between the two fluids (Ivanov *et al.*, 1992), and also where intermediate agents are used (Peneva *et al.*, 1992).

The majority of all other papers published in the literature are based on the concept of pinch (Linnhoff *et al.*, 1982), modified in order to accommodate the complications introduced by the time-varying operation of batch processes. The most rudimentary approach (Linnhoff *et al.*, 1988) treats batch plants as pseudo-continuous ones, time-averaging the energy requirements of the various processing steps and applying a cascade analysis to the grand composite curve that arises from the pseudo-continuous plant. This "Time Average Model" (TAM) approach is useful in that it sets rigorous targets for minimum utility consumption, against which actual consumption can be compared.

Kemp and MacDonald (1988a, b) do take into account the time-varying nature of batch processes by assuming a fixed production schedule. Opportunities for heat exchange are identified by inspecting the energy requirements and temperatures of the processing steps. They utilise a "Time-Temperature Cascade" (TTC) approach, where both direct and indirect heat exchange is possible. The use of heat storage in the latter case means that heat flows can be both time and temperature cascaded. The authors assume that direct heat transfer is always preferable to indirect heat transfer, and are thus able to derive targets for both energy storage and utility consumption. However, the utility consumption targets are over-conservative as no rescheduling is undertaken, and tend to be very different to the minimum targets predicted by the TAM.

Obeng and Ashton (1988) consider a fixed sequence of operations, and identify heat exchange opportunities using a time-slice model (TSM), which determines the intervals over which only direct heat exchange may occur, but otherwise bears some resemblance to the TTC model. This is used not only to predict utility consumption targets, but also to identify opportunities for adjusting the timing of operations to reduce these targets. An example is presented to demonstrate that, with appropriate retiming of operations, these targets approach the absolute ones predicted by the TAM. Kemp and Deakin (1989) extend the work of Obeng and Ashton by considering both the direct and indirect modes of heat exchange as well as retiming of operations.

More recently, Corominas *et al.* (1993) considered the problem of designing a minimum cost heat exchanger network and a heat exchange strategy for multiproduct plants operating in campaign mode. A pre-specified campaign comprising product batches and hot and cold stream information is examined to determine the best hot/cold stream match according to rules aiming to achieve maximum heat exchange. Since the authors only consider direct heat transfer, it may be necessary to retime the operations slightly to ensure that the paired streams are available simultaneously. The matches selected imply the heat exchanger network design, thus allowing the campaign to be costed. Different campaigns may be proposed, costed and compared using this approach.

Despite its clear importance, the minimisation of the cost of external utilities consumed is not usually the primary objective in scheduling the operation of a multipurpose plant. This is a consequence partly of the paramount demand for timely satisfaction of the multiple production requirements imposed on these plants, and partly of the often small proportion of energy costs compared to the high value of the raw materials and products produced in many such plants (*e.g.* in the pharmaceutical industry). It could, therefore, be argued that optimising the exploitation of any heat integration opportunities afforded by a *fixed* production schedule that already achieves all other plant objectives is indeed a reasonable approach.

However, in general, even optimal production schedules tend to be quite degenerate, in the sense that there often exist a large number of different schedules, all of which can achieve a given set of production requirements. However, the potential for heat integration could vary significantly from one such schedule to another. Furthermore, in some industrial sectors (*e.g.* food, dairy, brewing) that employ multipurpose plants, energy costs *do* form a significant proportion of the total production cost, and thus have to be balanced properly against other costs, such as those of the raw materials and manpower, and the value of the products (see, *e.g.*, Mignon and Hermia, 1993).

6.2 Overview of Proposed Approach

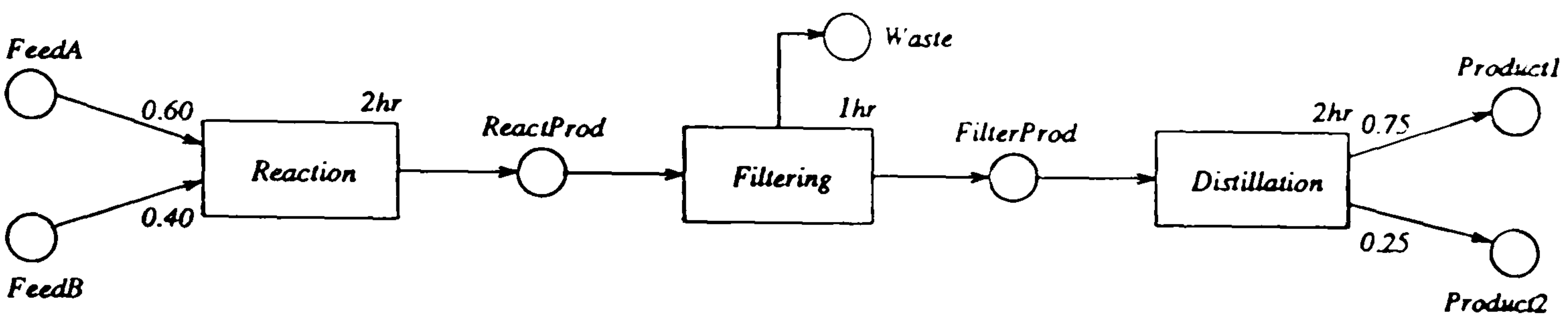
We consider two potential modes of energy integration. In *direct* energy integration, heat exchange occurs between two processing steps taking place *simultaneously* in two separate items of equipment coupled by a heat exchanger, and therefore a degree of time overlap of these operations must be ensured. On the other hand, *indirect* energy integration utilises a heat transfer medium (HTM) which acts as a mechanism both for transferring heat from one operation to another and for storing energy over time. This provides additional operational flexibility as it decouples, to a certain extent, the need for synchronisation of the various processing steps involved.

This chapter demonstrates how energy integration aspects can be incorporated within a general mathematical formulation for the scheduling of multipurpose plant operation (Kondili *et al.*, 1993), thus ensuring that all the relevant scheduling aspects are taken into account. More specifically, it is shown that this involves only relatively minor modifications to existing detailed scheduling formulations for the direct heat integration case. However, the mathematical formulation for the *indirect* heat integration case is based on a detailed characterisation of the variation of the mass and energy holdups of HTM over time. In particular, it takes account of the limitations on energy storage due to heat losses to the environment. A modified branch-and-bound procedure is proposed for the solution of the resulting nonconvex mixed integer nonlinear programming problem (MINLP).

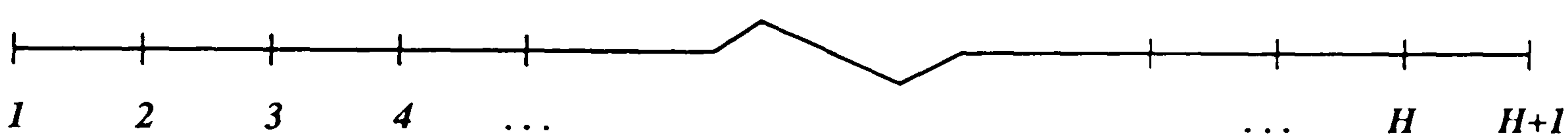
We consider a short-term scheduling problem that seeks to determine the optimal utilisation of the available plant resources (processing equipment, storage capacity, utility, manpower *etc.*) over a given time horizon. The plant may manufacture one or more products, and minimum and/or maximum production requirements may be imposed on them at various times during, or at the end of the horizon. Assuming that the problem posed is feasible, the objective is to select the schedule that maximises an economic performance measure taking into account the value of the products, and the cost of raw materials and utilities.

6.3 Basis of Mathematical Formulation

The mathematical formulations presented in this chapter are extensions of the general formulation of the short-term production scheduling problem presented by Kondili *et al.* (1993). The key concepts of this formulation, namely the STN process representation and the uniform discretisation of time are shown in figure 6.1.



(a) State-Task Network Process Representation



(b) Uniform Discretisation of the Time Horizon

Figure 6.1: Key Concepts of Kondili et al. (1993) Formulation

The time horizon is discretised into a number of intervals of equal duration (see figure 6.1b). System events (task starts and finishes, changes in resource availability, product demands *etc.*) are allowed to occur only at the boundaries of time intervals. The key variables of the formulation characterise the utilisation of resources over the time horizon, as follows:

I. Utilisation of Processing Equipment.

(a) Equipment Allocation Variables.

W_{ijt} = 1 if equipment unit j starts processing a batch of task i at time t; 0 otherwise.

(b) Batchsize variables.

B_{ijt} = amount of material that starts undergoing task i in unit j at time t.

II. Utilisation of Dedicated Storage Capacity.

S_{st} = amount of material in state s held in storage over time interval t.

III. Utilisation of Utilities and Manpower.

U_{ut} = amount of utility u being used over time interval t.

The above variables are subject to a number of constraints, including

(a) processing equipment allocation constraints expressing the fact that multipurpose equipment can carry out at most one task over any given time interval;

(b) processing and storage equipment capacity constraints;

(c) utility utilisation and availability constraints; and

(d) mass balance constraints describing the variation of the inventory of each material over time.

The precise mathematical form of these constraints is not important for the purposes of this chapter.

Finally, the objective function is a measure of the economic performance of the plant over the available time horizon, including the value of the products, the cost of the raw materials and the cost of utilities. In a somewhat simplified form, it can be written as

$$\max \sum_s C_s(S_{s,H+1} - S_{s,0}) - \sum_u \sum_{t=1}^{H} C_{ut} U_{ut} \tag{6.1}$$

The difference $(S_{s,H+1} - S_{s,0})$ in the first term reflects the nett increase in the inventory of material s over the time horizon, and this is multiplied by the unit value of this material, C_s. The second term represents the cost of the utilities used by the plant over the horizon, with C_{ut} being the unit cost of utility u over interval t.

The inclusion of utility costs in the objective function (6.1) makes this formulation suitable for considering opportunities for heat integration on a firm economic basis that takes direct account of the trade-offs between energy savings on one hand, and increased production and satisfaction of scheduling constraints (such as timely delivery of orders) on the other.

The next two sections consider the cases of direct and indirect energy integration respectively.

6.4 Direct Energy Integration

6.4.1 Problem Characteristics

We assume that the processing equipment available in the plant includes at least one pair of units (*e.g.* a reactor and a batch still) which are coupled to each other through a heat exchanger. This permits heat exchange to take place between two processing steps (*e.g.* an exothermic reaction and a distillation operation) that are performed simultaneously in these units.

We further assume that the precise operating mode and characteristics of two heat-integrated processing steps taking place in a given pair of equipment items are known *a priori*. This information includes the processing times of each of the two steps, as well as a fixed offset between their starting times. Such an offset may be necessary in order to maximise the potential for heat integration. For instance, in the case of heat-integrated reaction/distillation operations, it may be appropriate to start the reaction step earlier than the distillation step. This would allow sufficient time for the reaction to reach the operating temperature at which the heat consumed by the distillation is to be generated. In chapter 7, we describe a procedure for determining optimal policies for thermally coupled batch operations using dynamic optimisation techniques.

In general, the instantaneous rates of heat production and consumption of the two coupled tasks may be different in magnitude. We therefore assume that the equipment is also fitted with heat exchange facilities that allow the use of external utilities (*e.g.* steam, cooling water) to supplement the heating and cooling loads provided through heat integration. The existence of these additional facilities also makes possible the use of each of the equipment items in a pair for carrying out the corresponding processing step separately, if necessary, *i.e.* without the need for heat integration. Of course, the processing times of the individual steps when performed separately may well be different to those for the same steps when heat integration is employed. In any case, the instantaneous rates of consumption of external utilities over the duration of each step are assumed to be known, functions of the batchsize undergoing processing.

6.4.2 Mathematical Formulation for the Direct Heat Integration Case

The additional complications introduced by direct heat integration can be accommodated through a few simple extensions of the basic scheduling formulation. Each of the allowed heat exchange matches is represented by a "triplet" comprising two pairs of tasks and units, and an offset with respect to their starting times, as illustrated in figure 6.2.

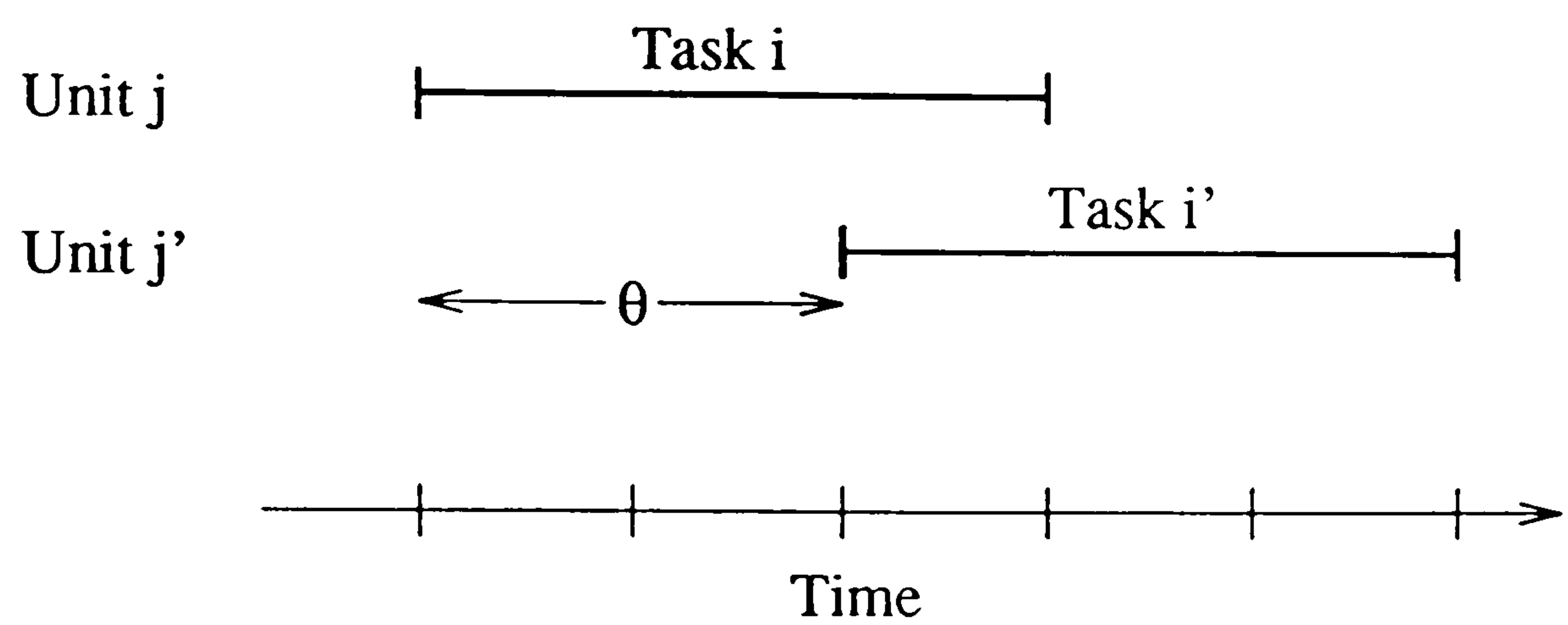


Figure 6.2: Triplet $\{(i, j), (i', j'), \theta\}$

Thus, the triplet $\{(i, j), (i', j'), \theta\}$ represents the possibility of exchanging heat between task i when performed in unit j, and task i' when performed in unit j', with a time offset of θ time intervals between their corresponding starting times. A given pair of processing units (j, j') coupled by a heat exchanger may be used for carrying out more than one pair of thermally-coupled processing steps, but each such possibility will be represented by a separate triplet.

The first issue that needs to be addressed is allowing for the possibility of each of the operations (i, i') actually taking place *without* any heat integration by relying exclusively on the use of external utilities. To achieve this, we split each of the original STN tasks i, i' into two, one corresponding to the heat-integrated operation of the task, and one to the stand-alone operation, as shown in figure 6.3.

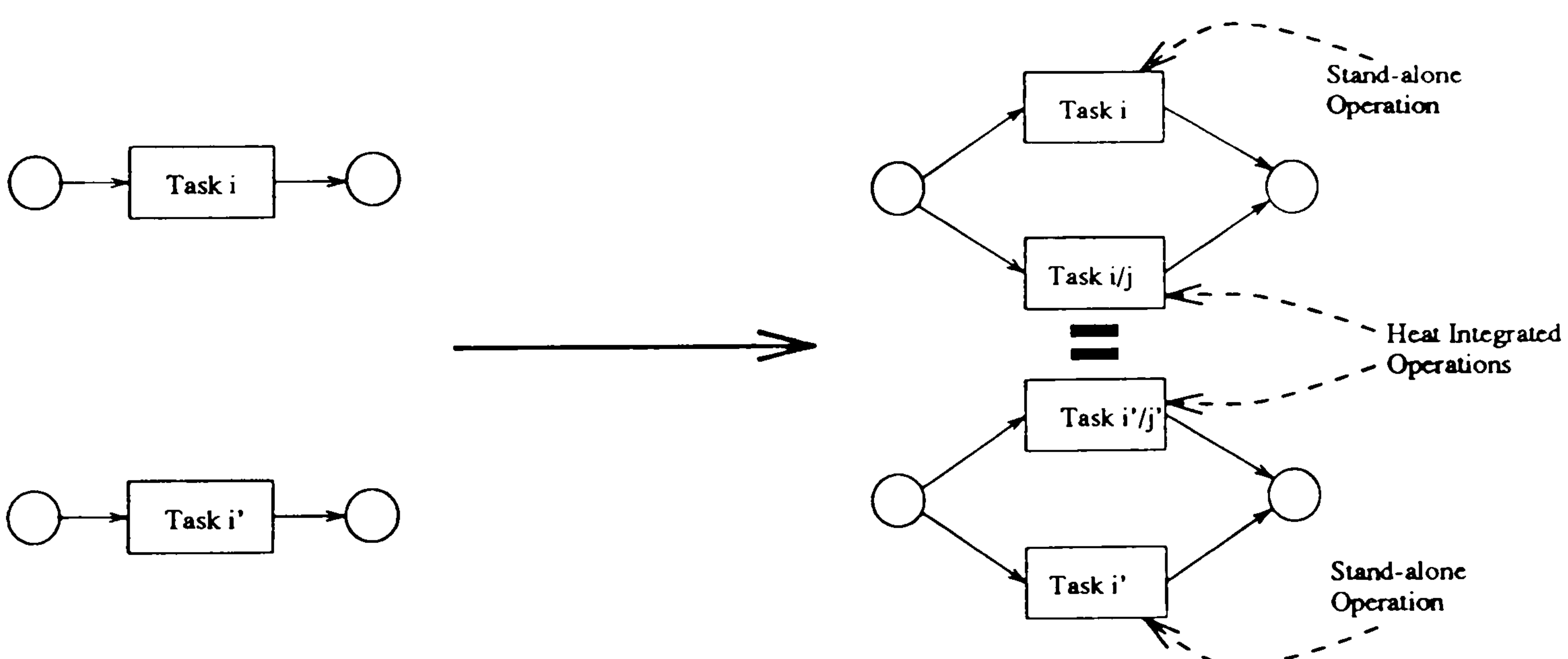


Figure 6.3: STN Modification for Modelling Heat-integrated Operations

Here the notation i/j is used to denote task i performed specifically in vessel j with heat integration. It should be noted that both the stand-alone tasks and the heat-integrated ones involve the *same* transformations of material.

The second issue to be addressed is that of ensuring that the heat-integrated tasks i/j and i'/j' always start with the specified time offset θ. This can be achieved by introducing constraints relating the corresponding equipment allocation variables in the scheduling formulation. In particular, we need to ensure that variables $W_{i/j,jt}$ and $W_{i'/j',j',t+\theta}$ have the same value, *i.e.* a batch of task i/j starts in unit j at time t if and only if a batch of task i'/j' starts in unit j' at time $t+\theta$:

$$W_{i/j,jt} = W_{i'/j',j',t+\theta} \tag{6.2}$$

Alternatively, the number of binary variables and constraints can be reduced by using (6.2) to eliminate half of the binary variables involved, thus introducing only one binary variable for each triplet $\{(i, j), (i', j'), \theta\}$ at each time instance t.

Finally, the ratio of the batchsizes of the heat integrated tasks will usually have to lie within given limits in order for their combined operation to be feasible. This requirement can be expressed mathematically as

$$\mu^{\min}_{ii'jj'} B_{i'/j',j',t+\theta} \le B_{i/j,jt} \le \mu^{\max}_{ii'jj'} B_{i'/j',j',t+\theta} \tag{6.3}$$

where $\mu^{\min}_{ii'jj'}$ and $\mu^{\max}_{ii'jj'}$ represent lower and upper bounds on the ratio of the batchsizes of the two heat integrated tasks.

6.4.3 Example of Direct Energy Integration

We consider a plant manufacturing two products, *Product1* and *Product2*, according to the following recipe which corresponds to the STN of figure 6.1a.

Reaction: Allow a mixture of 60% of *FeedA* and 40% of *FeedB* to react for 2 hours in order to produce intermediate *ReactProd*. This is an exothermic reaction requiring cooling throughout its duration. For a batchsize of B *te*, a constant cooling water flowrate of $1.59 + 0.10B$ *te/hr* is needed throughout the reaction.

Filtering: Filter intermediate *ReactProd* for 1 hour to form intermediate *FilterProd*. A small amount of solid waste is also produced at this step, but this can be ignored for the purposes of this example.

Distillation: Distill intermediate *FilterProd* to produce 75% of *Product1* and 25% of *Product2* after 2 hours. For a batchsize of B *te*, a constant steam supply to the reboiler of $0.044 + 0.0035B$ *te/hr* is required throughout the operation.

The available processing equipment comprises a 60 *te* reactor, a 80 *te* filter and a 70 *te* batch distillation column. The minimum allowable size of batches being processed in the reactor and the column cannot be less than 25% of their nominal capacity. The corresponding figure for the filter is 10%. Sufficient dedicated storage is provided for all raw materials and final products, whilst storage vessels of 100 *te* are available for each of the two intermediates. Unlimited availability of steam and cooling water is assumed, with corresponding unit costs of 200 and 4 relative cost units (*rcu*) per tonne respectively.

The time horizon of interest is 48 hours. The unit values for the two products are assumed to be equal at 5 *rcu/te*. The objective function is the maximisation of the combined value of the production over the time horizon, incorporating both the value of the products produced and the cost of the external utilities (steam and cooling water) consumed.

Assuming appropriate temperature levels, there is in principle the opportunity of exchanging heat between the *Reaction* task which requires cooling, and the *Distillation* task which requires heating. An equipment configuration that would permit this heat exchange to be realised is illustrated in figure 6.4.

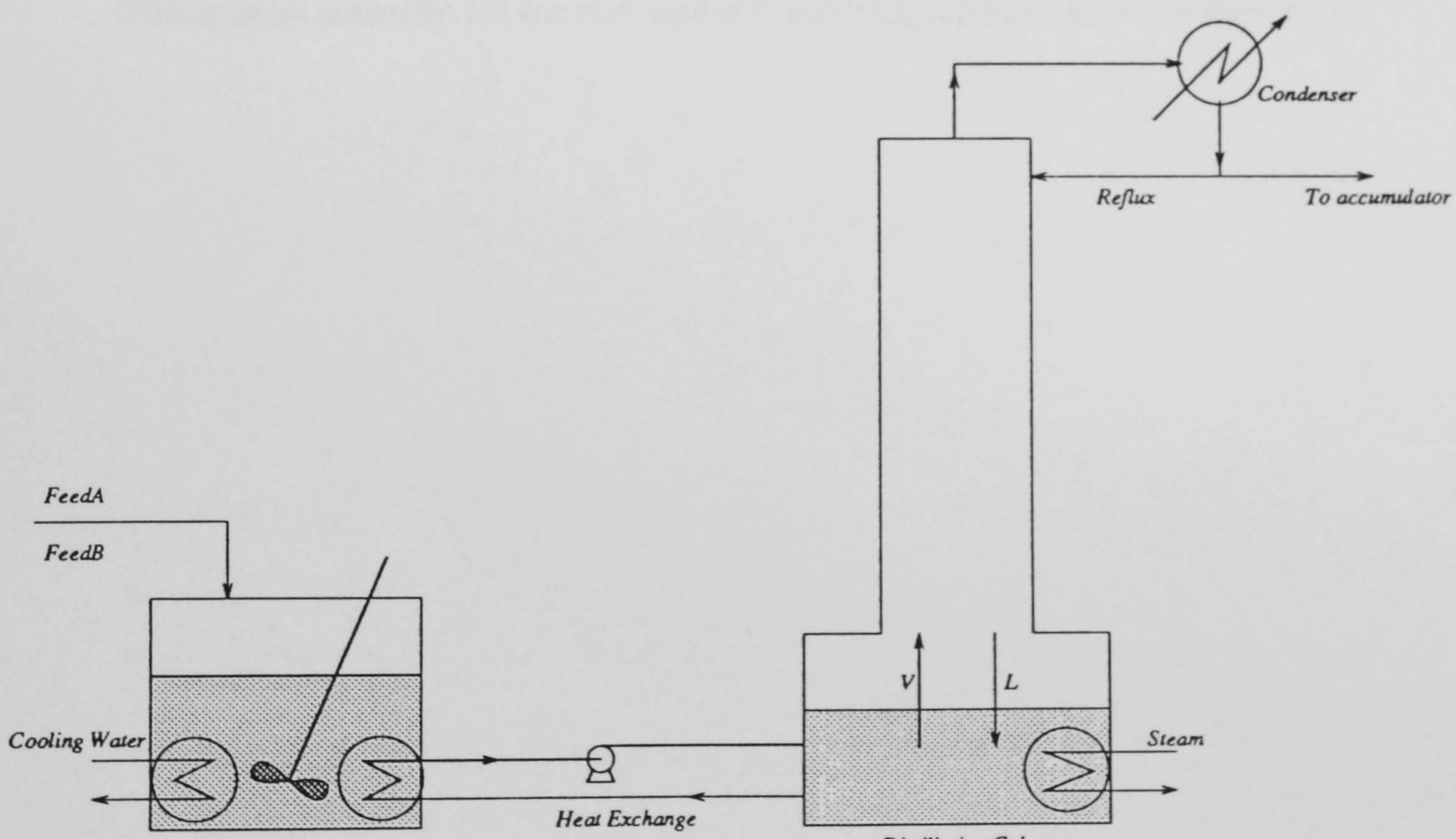


Figure 6.4: Direct Heat Exchange Network for Example Process

More specifically, this heat exchange match would involve task *Reaction* being performed in unit *Reactor* and the *Distillation* task in unit *Column* with an offset of 1 hour with respect to their starting times. Due to lower temperature differences, the duration of the reaction when operated in heat-integrated mode is increased from 2 to 3 hours.

As expected, the demands posed by these tasks on external utilities are also modified. Thus, for a batchsize of B *te*, the heat-integrated reaction task requires a cooling water flowrate of $1.0+0.06B$ *te/hr* during the first hour of its operation only; thereafter all its needs are satisfied through heat exchange with the contents of the distillation column reboiler.

On the other hand, the heat-integrated distillation task requires a constant steam flowrate of $0.020 + 0.0016B$ *te/hr* throughout its duration supplementing the energy received from the heat exchange with the reactor contents. The STN for the process taking into account both stand-alone and heat-integrated operations is shown in figure 6.5.

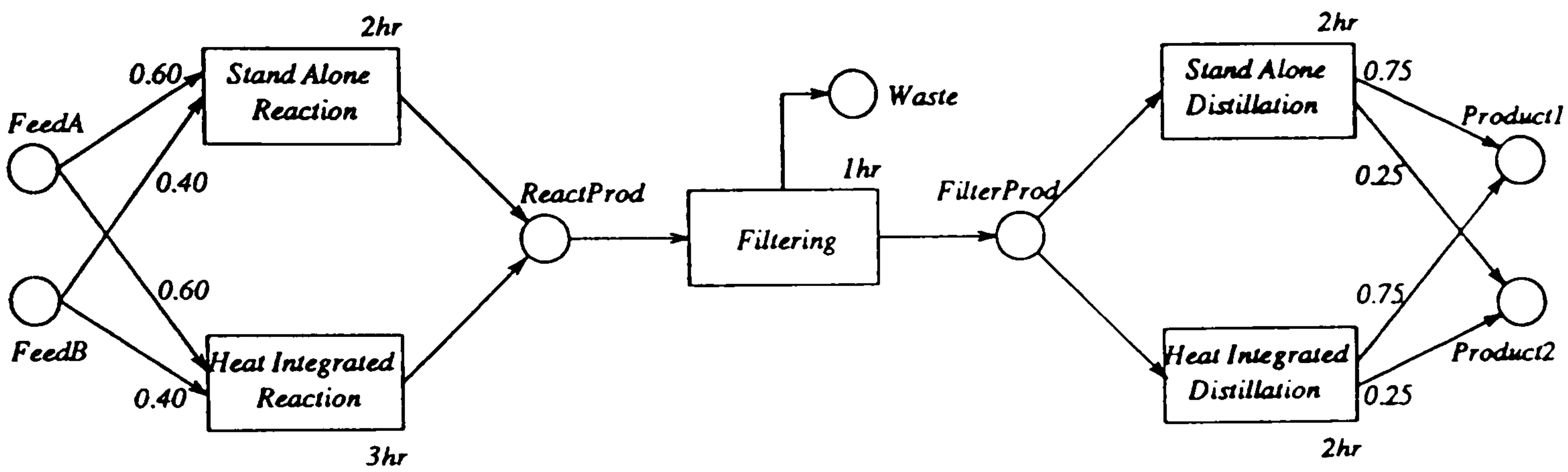
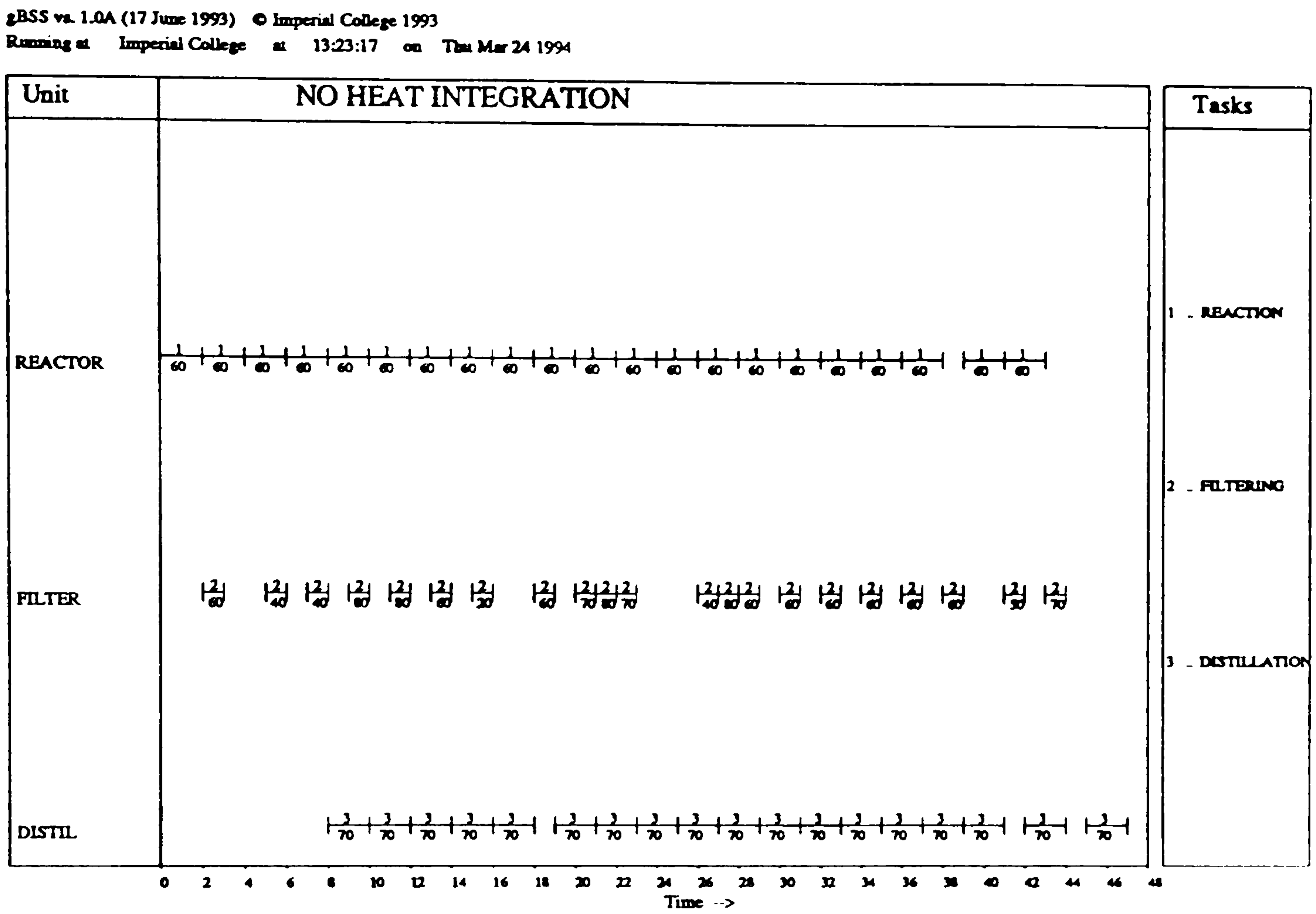
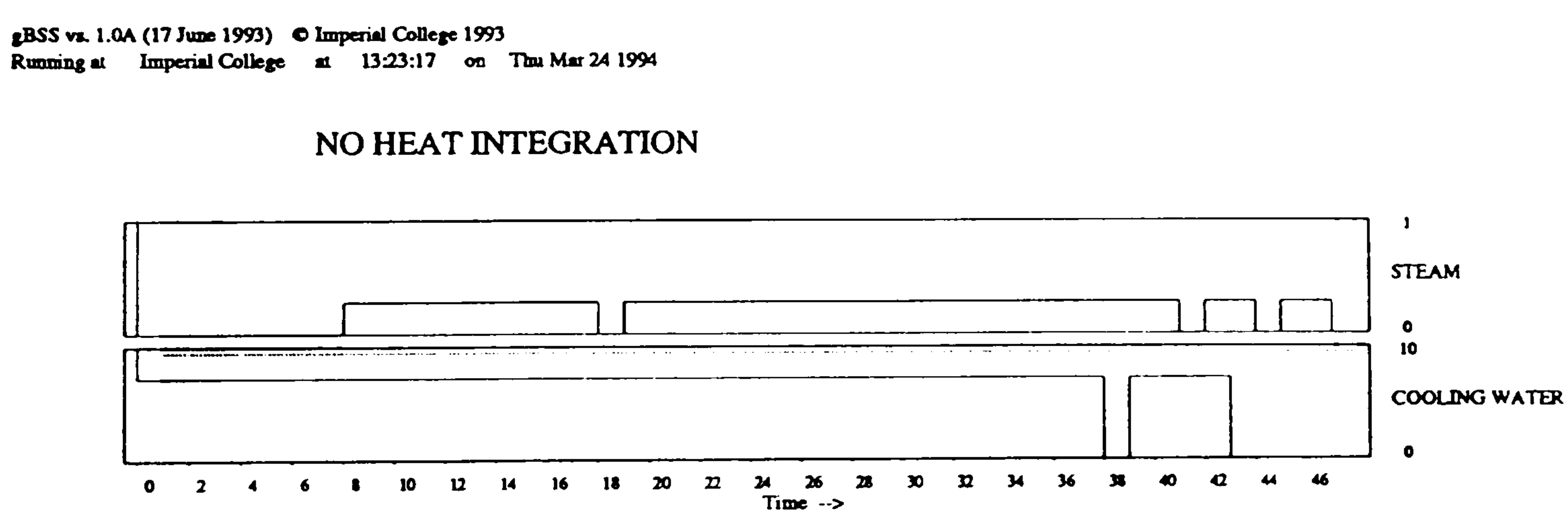


Figure 6.5: Modified STN for Example Process

The optimal schedule for the case *without* heat integration is shown in figure 6.6.



(a) Utilisation of Processing Equipment

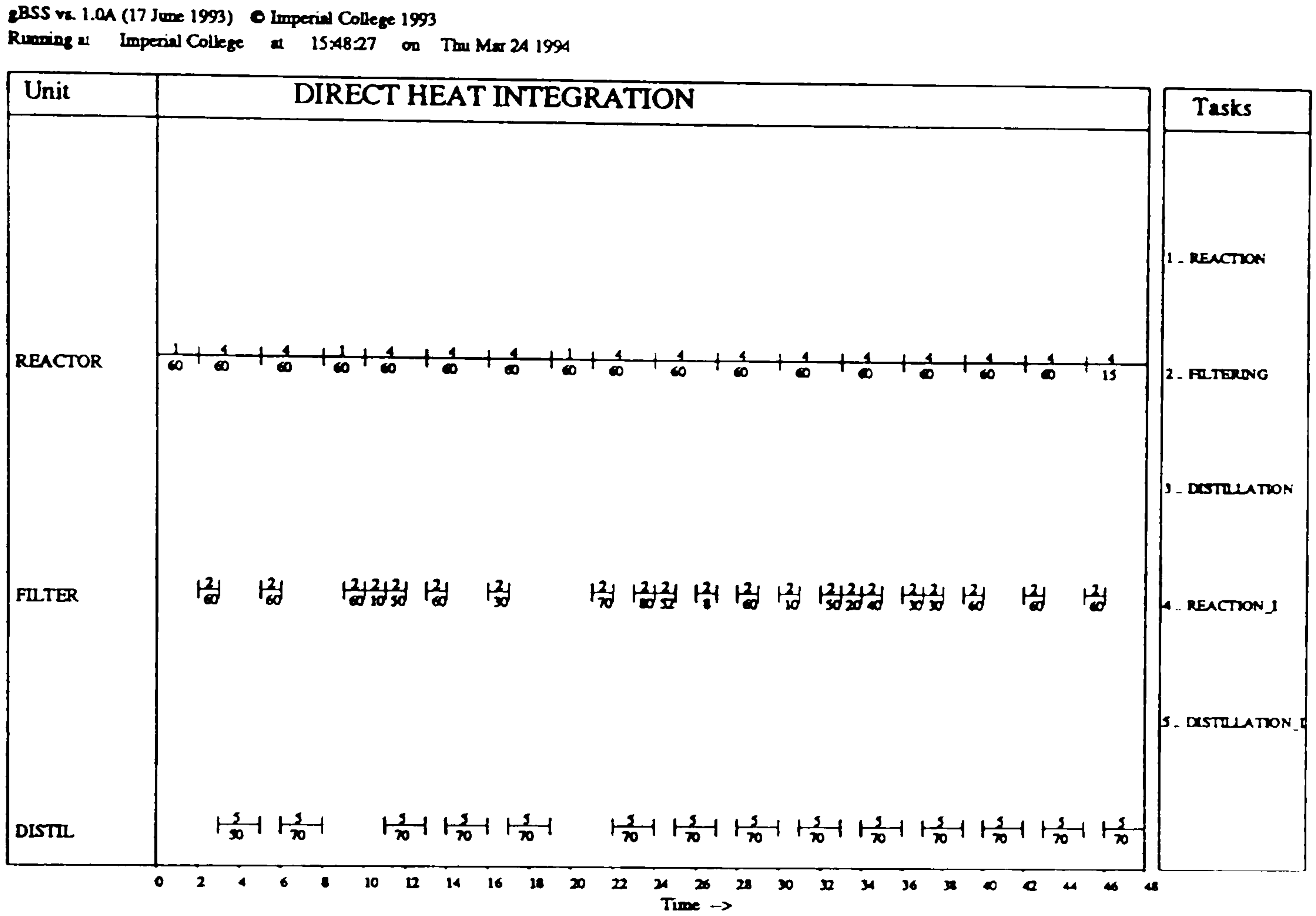


(b) Utilisation of External Utilities

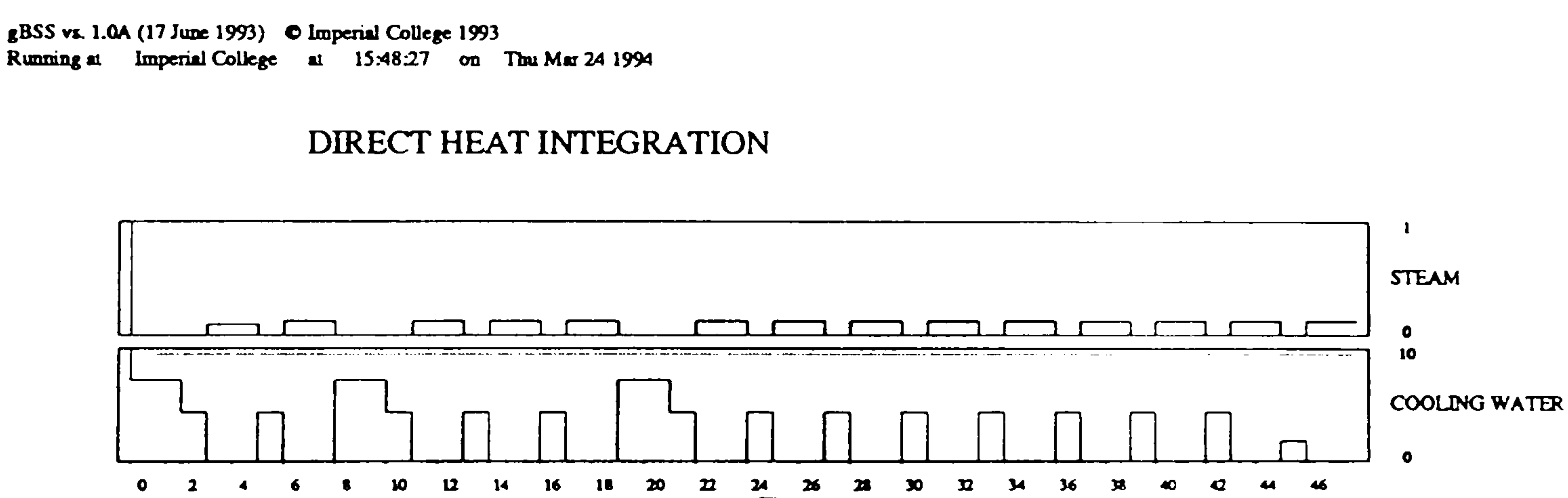
Figure 6.6: Optimal Schedule without Heat Integration

The number above each line segment in the Gantt chart denotes the task being carried out (see key in right hand side margin of chart). The number below the segment is the amount of material being processed (in *te*). The amounts of the two products produced are 945 and 315 *te* respectively, while the costs of steam and cooling water consumption are 2080.8 *rcu* and 1275.1 *rcu*. Overall, this results in an objective function value of 2944.1 *rcu*.

Figure 6.7 shows the optimal schedule for the case where heat integration *is* allowed.



(a) Utilisation of Processing Equipment



(b) Utilisation of External Utilities

Figure 6.7: Optimal Schedule with Direct Heat Integration

It can be seen that this involves both stand-alone operations and integrated ones. In particular, a batch of the stand-alone *Reaction* (task 1) has to be performed at the start of the horizon, a period during which the integrated operation (tasks 4 and 5) cannot take place due to the unavailability of intermediate *FilterProd* feeding the *Distillation* task. More interestingly, stand-alone reaction tasks are also performed sporadically throughout the horizon. This feature of the optimal solution is due to the complex trade-off between increased processing time and reduced utility consumption for the reaction task. On the other hand, since heat integration leaves the processing time of the distillation unaltered, no stand-alone distillation operations take place.

The amounts of the two products produced in this second case are 720 and 240 *te* respectively, which are lower than the corresponding figures for the case without heat integration. This is to be expected as the heat integrated reaction task takes longer than its stand-alone counterpart. On the other hand, the costs of the utilities consumed are reduced to an even larger extent, down to 726.4 *rcu* of steam and 429.0 *rcu* of cooling water. Overall, this results in an objective function value of 3644.6 *rcu*, representing a 23.8% improvement over the case without heat integration.

The mathematical formulations of the two cases as MILPs involve 142 and 188 binary variables respectively. The optimal solutions were obtained after the examination of 17 and 5 nodes of the search tree respectively. A 5% margin of optimality was used for the branch-and-bound procedure. The optimal solutions were respectively within 4.76% and 0.37% of the objective function values of the corresponding fully relaxed linear programmes.

6.5 Indirect Energy Integration

6.5.1 Problem Characteristics

The indirect mode of heat exchange involves the use of a heat transfer medium (HTM) which is assumed to be generally available at a number of different temperature levels, each kept in a separate, well-mixed, and possibly insulated storage tank. The existence of storage capacity for the HTM effectively removes the requirement for strict synchronisation between the tasks producing and those consuming heat, thus permitting more flexibility in plant operation.

6.5.1.1 Network Connectivity

As shown schematically in figure 6.8, it is assumed that direct pipe connections exist between the processing units in which the tasks take place on one hand, and the HTM storage tanks on the other. This allows each unit to be used independently of all others in the network, provided of course that sufficient amounts of HTM are available at the required temperature(s).

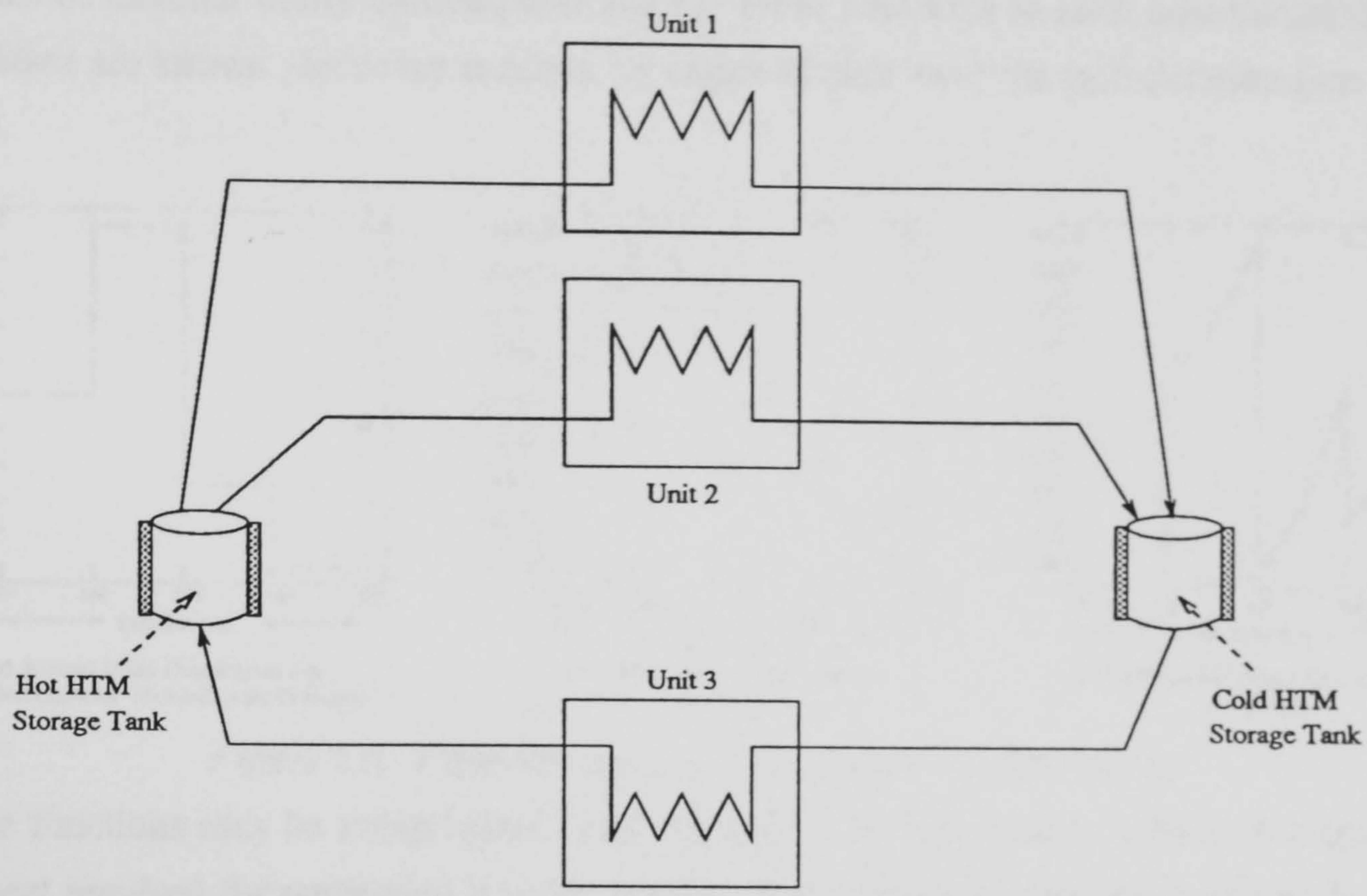


Figure 6.8: Heat Transfer Network for Indirect Heat Integration

Each processing unit may be connected to more than one HTM storage tank. However, we assume that only one of its inlet and one of the outlet connections are utilised during any one heat-integrated task taking place in a given unit.

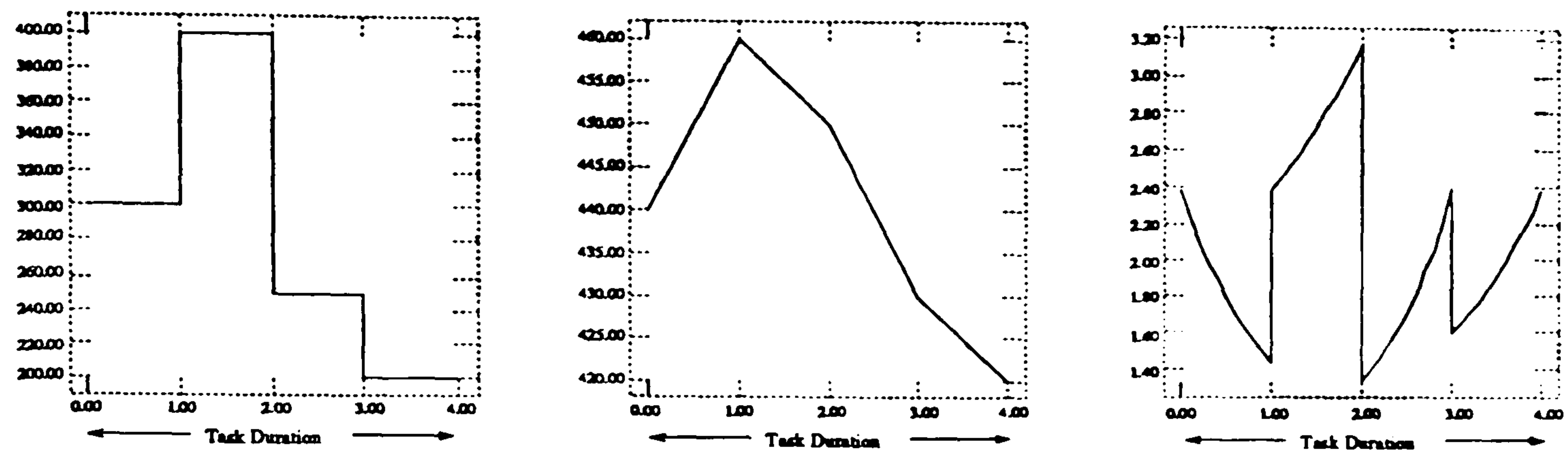
As for the direct heat integration mode, it is also assumed that the processing units in the heat transfer network are equipped with facilities allowing the use of external utilities, either supplementing or replacing the use of the HTM.

6.5.1.2 Operating Policies for Heat-integrated Operations

The operating policy for a batch operation can be expressed in terms of its duration and the variation of the available control variables (*e.g.* the reflux ratio in a batch distillation column, or the rate of steam supply to its reboiler) over this duration. For heat-integrated batch operations, the heat load provided by the heat integration can be viewed as an additional manipulated variable.

The above operating decision variables may be specified in an *ad hoc* manner aiming to satisfy various end-point constraints (*e.g.* purity of the final products) or path constraints (*e.g.* maximum temperature throughout the operation). Alternatively, they may be chosen so as to optimise some performance criterion (*e.g.* maximise yield) subject to the same constraints. This usually involves the solution of complex dynamic optimisation problems.

For the purposes of this chapter, we assume that the operating policy for each operation (heat-integrated or otherwise) is fixed and known *a priori*. We further assume that the time profiles of external utility consumption and the HTM heat load in each heat-integrated unit operation are known piecewise constant functions of time over the task duration (see figure 6.9a).



(a) External Utility Consumption and HTM Heat Loads for Heat-Integrated Operations

(b) Temperature in Heat-Integrated Operations

(c) HTM Flowrate through Heat-Integrated Operations

Figure 6.9: Characteristics of Heat-integrated Operations

These functions may be independent of the batchsize, or vary linearly with it. For instance, the heat required for preheating a metal reactor vessel before loading the reactants is effectively determined by the mass of the vessel. On the other hand, the cooling or heating load demanded by the reactor contents during the course of the reaction is essentially proportional to the amount of material undergoing processing.

Once the duration and operating policy for a heat integrated operation are fixed, the time variation of the temperature of the material undergoing the operation can also be determined, either by simulation or experimentally. In general, this variation will be a known continuous function of time, of the form shown in figure 6.9b.

The overall plant schedule must ensure that any heat-integrated operation taking place receives both the required external utilities and the necessary heat load from the HTM at the correct levels throughout its duration. This is not entirely straighforward because the temperature of the stored HTM may vary over the time horizon of interest, which in turn affects the temperature difference that drives the heat transfer between the unit operation and the HTM flowing through it. Thus we need to determine suitable time profiles for the flowrates of the HTM. As will be explained later, in general these may be discontinuous functions of time over the duration of the task (see figure 6.9c).

6.5.1.3 HTM Storage

Unless the HTM storage vessels are heavily insulated and/or the corresponding HTM temperatures are very close to ambient levels, some heat losses (or gains) to (or from) the environment will be unavoidable. Depending on the thermal conductivity of the material of construction of the storage vessel, and also the geometry of the latter, the effective heat transfer area for these interactions with the environment may be constant or it may depend on the amount of material held in the tank at any given time.

The other important factor in determining the rate of heat loss or gain is the ambient temperature, which may vary over the time horizon of interest. We assume that this variation is known *a priori*, and that the overall coefficient for heat transfer between each storage tank and the environment is constant and known.

Because of the heat losses or gains mentioned above, and also due to differences in the actual temperature at which HTM is produced by different tasks, the temperature of the HTM in each tank will generally vary over the time horizon of interest. However, in the interests of proper operation of the heat-integrated tasks, it is usually desirable that these temperatures be maintained within given upper and lower bounds. In practice, the main mechanism for achieving this is the adjustment of the timings of the various tasks taking place over the horizon as determined by the operating schedule, and the flowrates of the HTM used by each task.

6.5.2 Mathematical Formulation for the Indirect Heat Integration Case

The operation of the heat integration network during the time horizon of interest can be described in terms of the following functions of time:

$S_s(t)$	:	Amount of material in HTM storage tank s at time t.
$\Theta_s(t)$	:	Temperature of material in HTM storage tank s at time t.
$q_{ij}(t)$	:	Heat load to the HTM from contents of unit j carrying out heat-integrated task i at time t.
$f_{ij}(t)$	:	Flowrate of HTM through unit j carrying out heat-integrated task i at time t.
$\Theta_{ij}^{out}(t)$	:	Temperature of HTM leaving unit j carrying out heat-integrated task i at time t.

The approach we follow is to carry out rigorous mass and energy balances on each HTM storage tank and each heat-integrated operation. These lead to differential equations in time. We then integrate these equations over each discrete time interval, using trapezoidal approximations to the various integrals arising from the time integration.

The resulting discrete time formulation is expressed in terms of finite-dimensional sets of parameters corresponding to the values of the above variables at the time interval boundaries. As shown later in this section, the flowrates $f_{ij}(t)$ and the temperatures $\Theta_{ij}^{out}(t)$ are *not* necessarily continuous at the interval boundaries. We therefore need to introduce separate parameters representing the values of these variables at the end of one interval and the start of the next.

The discretisation parameters to be determined by the scheduling algorithm are therefore the following:

S_{st}	:	Amount of material in HTM storage tank s at time point t.
Θ_{st}	:	Temperature of material in HTM storage tank s at time point t.
q_{ijt}	:	Heat load to the HTM from contents of unit j carrying out heat-integrated task i over time interval t.
f_{ijt}	:	Flowrate of HTM through unit j carrying out heat-integrated task i at the start of time interval t.
f'_{ijt}	:	Flowrate of HTM through unit j carrying out heat-integrated task i at the end of time interval t.
Θ_{ijt}^{out}	:	Temperature of HTM leaving unit j carrying out heat-integrated task i at the start of time interval t.
Θ'^{out}_{ijt}	:	Temperature of HTM leaving unit j carrying out heat-integrated task i at the end of time interval t.

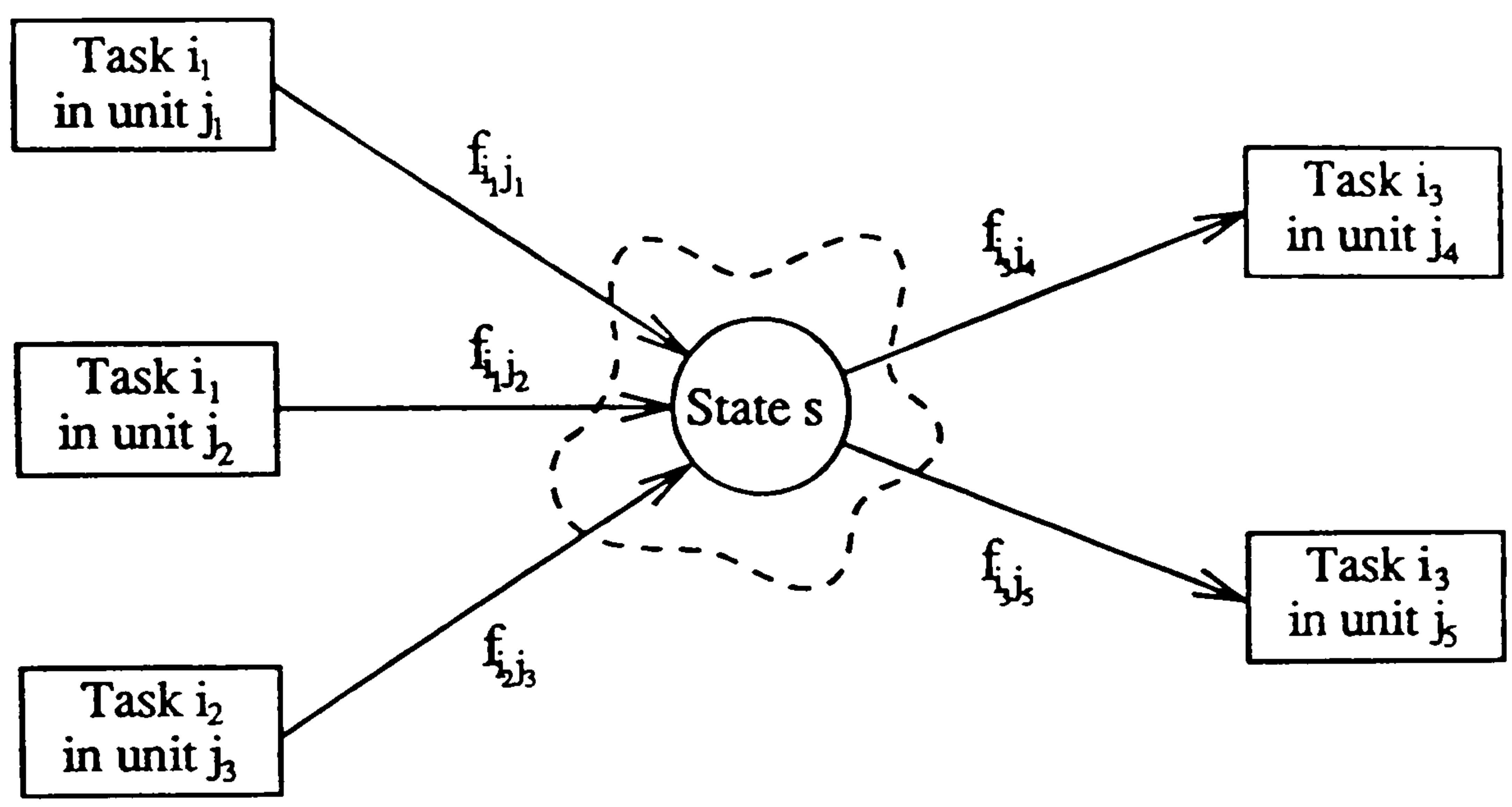
In addition to the above, the formulation also involves the variables of the original scheduling formulation described in section 6.3.

6.5.2.1 STN Modification

As in the case of direct heat integration, we start by modifying the process STN to take account of the possibility of the same task taking place either in a stand-alone or in a heat-integrated mode. This can be achieved by splitting each original task into two, both involving the same transformations of material, but otherwise being different with respect to their processing times and/or consumption of external utilities (*cf.* figures 6.3, 6.5).

6.5.2.2 Mass Balances on HTM Storage Tanks

As illustrated in figure 6.10, HTM in storage tank s is produced and consumed by subsets $\bar{T}_s$ and T_s respectively of the heat-integrated tasks in the network. In general, each task i may take place in more than one processing unit j at the same time.



$\overline{T}_s = \{i_1, i_2\}$ $\qquad$ $T_s = \{i_3\}$

$K_{i_1} = \{j_1, j_2\}$ $\qquad$ $K_{i_3} = \{j_4, j_5\}$

$K_{i_2} = \{j_3\}$

Figure 6.10: Mass Balance on HTM Storage Tank s

A mass balance in the continuous time domain yields the differential equation

$$\frac{dS_s}{dt} = \sum_{i \in \overline{T}_s} \sum_{j \in K_i} f_{ij} - \sum_{i \in T_s} \sum_{j \in K_i} f_{ij} \qquad \forall\, s, t \tag{6.4}$$

where K_i is the set of processing units that are suitable for task i.

By integrating (6.4) over time interval t and using the trapezoidal rule to approximate the integrals of the flowrates, we obtain:

$$S_{s,t+1} = S_{st} + \frac{\delta}{2} \sum_{i \in \overline{T}_s} \sum_{j \in K_i} (f_{ijt} + f'_{ijt}) - \frac{\delta}{2} \sum_{i \in T_s} \sum_{j \in K_i} (f_{ijt} + f'_{ijt}) \qquad \forall\, s, t \tag{6.4'}$$

where δ is the length of the discretisation interval.

6.5.2.3 Energy Balances on Heat-integrated Operations

We now consider the energy balance on a heat-integrated operation i taking place in a unit j. This receives HTM from a storage tank, denoted by s_{ij}, and discharges it into another HTM storage tank, denoted by $\overline{s}_{ij}$ (see figure 6.11).

Task i in unit j

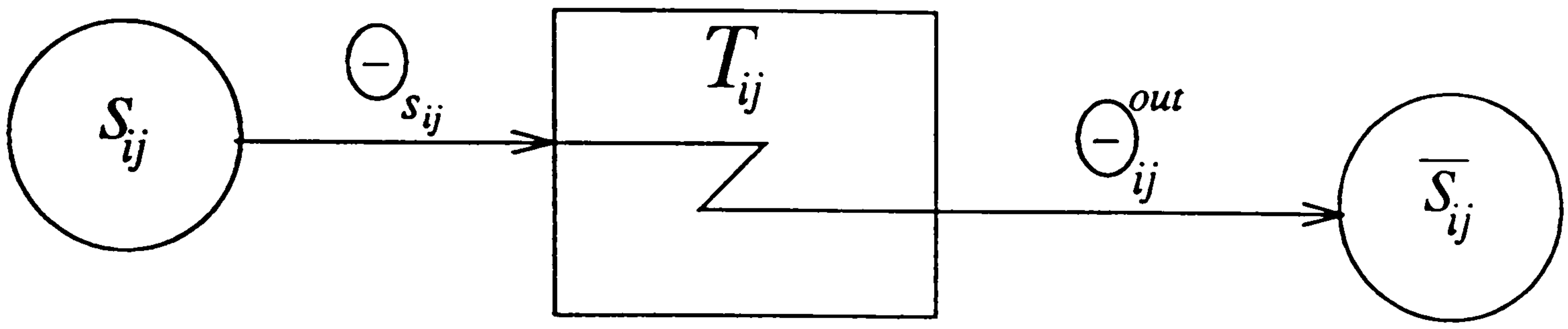


Figure 6.11: Energy Balance on Heat-integrated Operations

The temperatures of the HTM entering and leaving the unit at time t are denoted by $\Theta_{s_{ij}}(t)$ and $\Theta_{ij}^{out}(t)$, while the temperature of the fluid surrounding the heat transfer jacket (or coil) is assumed to be uniform and is denoted by $T_{ij}(t)$.

We assume that the dynamic behaviour of the jacket is fast compared to the rest of the operation, and that the rate of heat transfer $q_{ij}(t)$ between the fluid in the unit and the HTM flowing in the jacket can be described adequately in terms of an arithmetic mean temperature difference:

$$q_{ij}(t) = U_{ij} A_{ij} \left(T_{ij}(t) - \frac{\Theta_{s_{ij}}(t) + \Theta_{ij}^{out}(t)}{2} \right) \quad \forall\ i,\ j \in K_i,\ t \tag{6.5}$$

where U_{ij} and A_{ij} are the relevant heat transfer coefficient and area respectively.

An enthalpy balance on the HTM yields

$$q_{ij}(t) = f_{ij}(t) \left(h_{ij}(t) - h_{s_{ij}}(t) \right) \quad \forall\ i,\ j \in K_i,\ t \tag{6.6}$$

where $h_{s_{ij}}$ and h_{ij} are the specific enthalpies of the HTM in the supply storage tank and the HTM leaving the heat-integrated operation respectively. Assuming a constant specific heat capacity, c_p, for the HTM, we can write (6.6) as

$$q_{ij}(t) = f_{ij}(t)\ c_p \left(\Theta_{ij}^{out}(t) - \Theta_{s_{ij}}(t) \right) \quad \forall\ i,\ j \in K_i,\ t \tag{6.6'}$$

From (6.5) and (6.6'), we can eliminate the outlet temperature Θ_{ij}^{out} obtaining an expression for the flowrate f_{ij} in terms of the quantities q_{ij}, T_{ij} and $\Theta_{s_{ij}}$. Since the heat load q_{ij} is assumed to be a piecewise constant function of time while the temperatures T_{ij} and $\Theta_{s_{ij}}$ are continuous functions, we can deduce that $f_{ij}(t)$ will, in general, be a *discontinuous* function of time, of the form shown in figure 6.9c. The temperature $\Theta_{ij}^{out}(t)$ will be a similarly discontinuous function.

The temperature of the fluid surrounding the jacket at discrete time points θ ($\theta = 0..p_i$) relative to the start of heat-integrated task i taking place in unit j is also assumed to be known, and denoted by $\bar{T}_{ij\theta}$. The corresponding temperature in absolute time, T_{ijt} will depend on when the task actually started. This can be expressed as

$$T_{ijt} = \sum_{\theta=0}^{p_i-1} W_{ij,t-\theta}\bar{T}_{ij\theta} \qquad \forall\, i,\, j \in K_i,\, t \tag{6.7}$$

where p_i is the duration of task i. It should be noted that, because of the allocation constraints in the scheduling formulation (Kondili *et al.*, 1993), at most one of the binary $W_{ij,t-\theta}$ variables in the summation on the right hand side can be one. For instance, if the operation i started in unit j at time $t-2$, then $W_{ij,t-2} = 1$, and, assuming that $p_i > 2$, (6.7) will reduce correctly to $T_{ijt} = \bar{T}_{ij2}$. It may also be worth mentioning that (6.7) imply that $T_{ijt} = 0$ if operation i is ***not*** taking place in unit j at time t. This is acceptable as the function $T_{ij}(t)$ is, in fact, undefined at such times.

We can now produce discrete time versions of constraints (6.5) and (6.6) by writing them at the start and end of each interval and using (6.7) to eliminate the quantity T_{ijt}:

$$\begin{aligned}
q_{ijt} &= U_{ij} A_{ij} \left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta}\bar{T}_{ij\theta} - \frac{\Theta_{s_{ij}t} + \Theta_{ijt}^{out}}{2} \right) \\
&= U_{ij} A_{ij} \left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta}\bar{T}_{ij,\theta+1} - \frac{\Theta_{s_{ij},t+1} + \Theta'^{out}_{ijt}}{2} \right) \\
&= f_{ijt}\, c_p\, (\Theta_{ijt}^{out} - \Theta_{s_{ij}t}) \\
&= f'_{ijt}\, c_p\, (\Theta'^{out}_{ijt} - \Theta_{s_{ij},t+1}) \qquad \forall\, i,\, j \in K_i,\, t
\end{aligned} \tag{6.8}$$

The final task is to relate the heat load q_{ijt} to the amount of material undergoing the heat-integrated operation. Over time interval θ relative to the start of the operation, this heat load is given by

$$\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} B \qquad \theta = 0..p_i - 1 \tag{6.9}$$

where $\gamma_{ij\theta}^{(1)}$ and $\gamma_{ij\theta}^{(2)}$ are given constants and B is the batchsize.

When considering an interval t in ***absolute*** time, the heat transfer rate in a given (i, j) pair may be attributed to a batch of task i that started in unit j at time t, or to batches that started at earlier times $t - \theta$, $\theta = 1..p_i - 1$ and are still being processed (see figure 6.12).

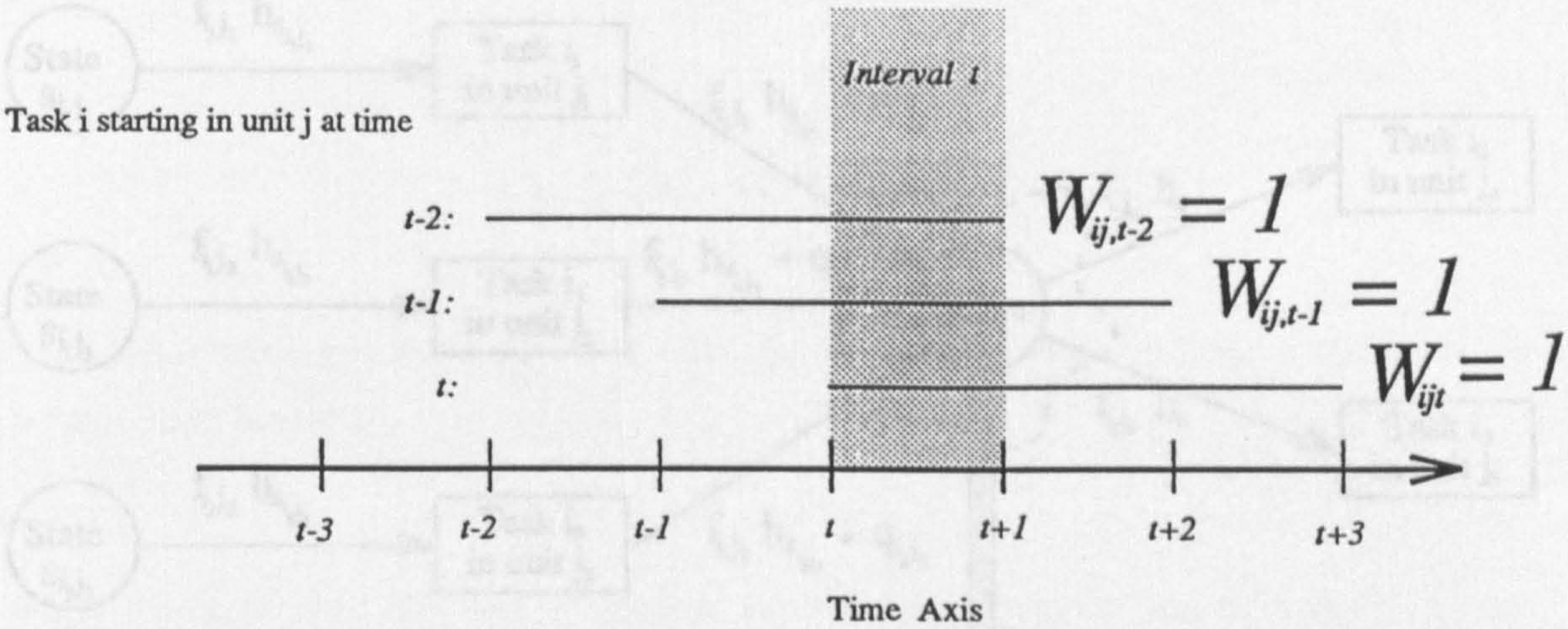


Figure 6:12: Heat Load Contributions over Interval t

To take this into account, we have to sum (6.9) over all possible θ, yielding

$$q_{ijt} = \sum_{\theta=0}^{p_i-1} (\gamma^{(1)}_{ij\theta} W_{ij,t-\theta} + \gamma^{(2)}_{ij\theta} B_{ij,t-\theta}) \quad \forall\, i,\, j \in K_i,\, t \tag{6.10}$$

The multiplication of $\gamma^{(1)}_{ij\theta}$ by the binary variable $W_{ij,t-\theta}$ is necessary to ensure that no contribution to the summation is made if task i has *not* started in unit j at time $t-\theta$ (corresponding to $W_{ij,t-\theta} = 0$).

6.5.2.4 Energy Balances on HTM Storage Tanks

An energy balance on the HTM storage tank s in the continuous time domain can be written (see figure 6.13) as

$$\frac{d(S_s u_s)}{dt} = \sum_{i \in \bar{T}_s} \sum_{j \in K_i} f_{ij} h_{ij} - \sum_{i \in T_s} \sum_{j \in K_i} f_{ij} h_s - U_s A_s(\Theta_s - T^{amb}) \quad \forall\, s,\, t \tag{6.11}$$

where $u_s(t)$ and $h_s(t)$ are respectively the specific internal energy and enthalpy of the contents of the tank, $T^{amb}(t)$ the ambient temperature, and U_s and A_s are respectively the heat transfer coefficient and the effective heat transfer area for heat losses to the environment. It should be noted that A_s may vary over time as it may depend on the amount of material held in the tank.

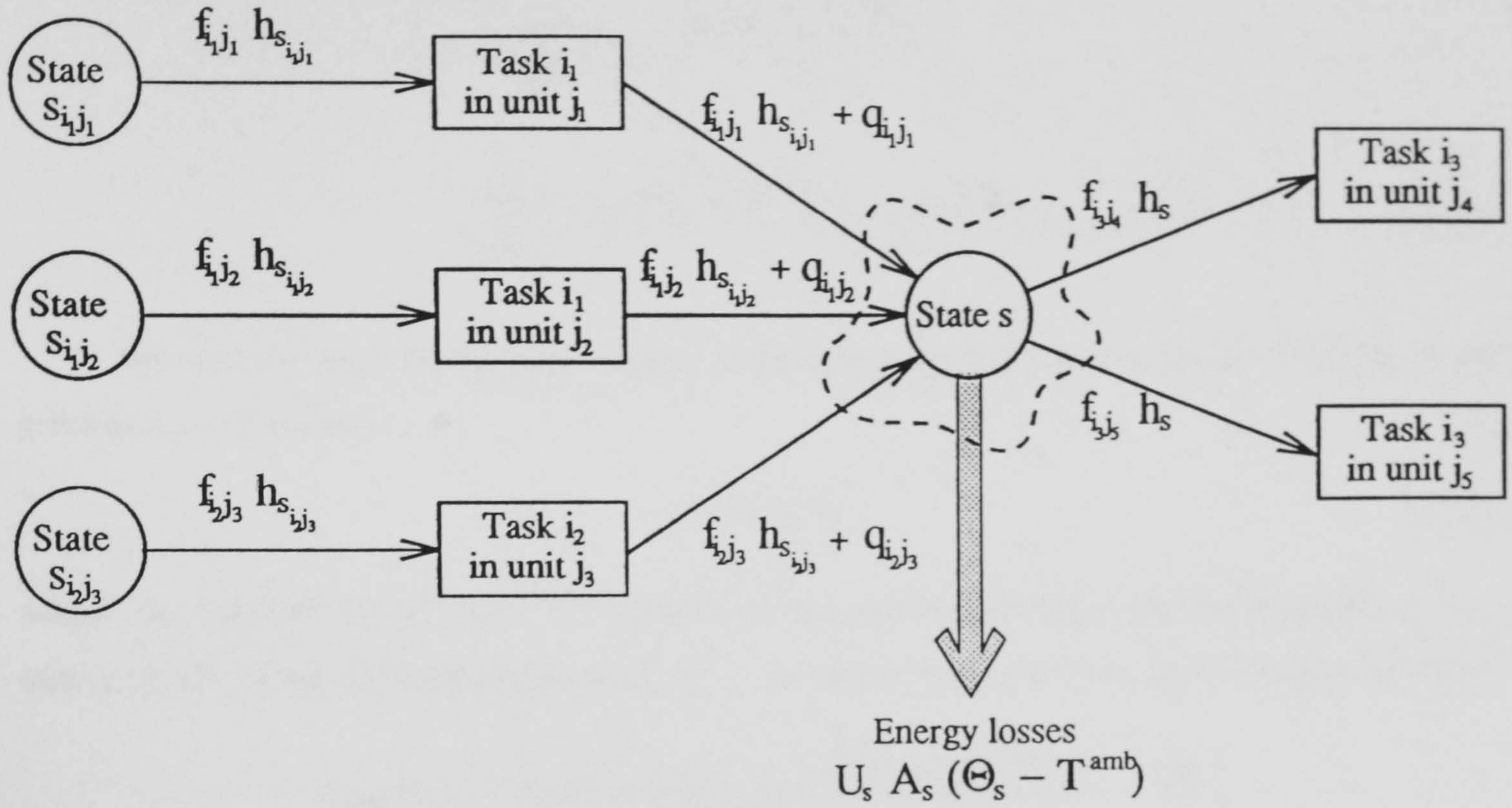


Figure 6.13: Energy Balance on HTM Storage Tank s

By combining (6.11) with (6.6), we can eliminate the specific enthalpy of the streams entering the tank, $h_{ij}(t)$, to obtain:

$$\frac{d(S_s u_s)}{dt} = \sum_{i\in\overline{T}_s} \sum_{j\in K_i} \left(f_{ij} h_{s_{ij}} + q_{ij} \right) - \sum_{i\in T_s} \sum_{j\in K_i} f_{ij} h_s - U_s A_s (\Theta_s - T^{amb}) \qquad \forall\, s, t \quad (6.11')$$

By integrating the above over time interval t, taking account of the fact that $q_{ij}(t)$ is constant during the interval, and replacing all integrals on the right hand side by trapezoidal approximations, we obtain the discrete time relation:

$$S_{s,t+1} u_{s,t+1} - S_{st} u_{st} = \delta \sum_{i\in\overline{T}_s} \sum_{j\in K_i} \left(\frac{f_{ijt} h_{s_{ij},t} + f'_{ijt} h_{s_{ij},t+1}}{2} + q_{ijt} \right)$$

$$- \delta \sum_{i\in T_s} \sum_{j\in K_i} \frac{f_{ijt} h_{st} + f'_{ijt} h_{s,t+1}}{2}$$

$$- U_s \delta \; \frac{A_{st}(\Theta_{st} - T_t^{amb}) + A_{s,t+1}(\Theta_{s,t+1} - T_{t+1}^{amb})}{2} \qquad \forall\, s, t \qquad (6.12)$$

For most HTMs of practical interest, the difference between internal energy and enthalpy is negligible. By further assuming constant specific heat capacity, we can rewrite (6.12) entirely in terms of temperatures, as follows:

$$S_{s,t+1} \Theta_{s,t+1} - S_{st} \Theta_{st} = \delta \sum_{i\in\overline{T}_s} \sum_{j\in K_i} \left(\frac{f_{ijt} \Theta_{s_{ij},t} + f'_{ijt} \Theta_{s_{ij},t+1}}{2} + \frac{q_{ijt}}{c_p} \right)$$

$$-\delta \sum_{i \in T_s} \sum_{j \in K_i} \frac{f_{ijt}\Theta_{st} + f'_{ijt}\Theta_{s,t+1}}{2}$$

$$-\frac{U_s\delta}{c_p} \frac{A_{st}(\Theta_{st} - T_t^{amb}) + A_{s,t+1}(\Theta_{s,t+1} - T_{t+1}^{amb})}{2} \quad \forall\, s, t \qquad (6.12')$$

We assume that, for a given vessel s, the area A_{st} is a simple linear function of the holdup S_{st} of the form

$$\alpha_s^{(1)} + \alpha_s^{(2)} S_{st} \qquad (6.13)$$

where the coefficients $\alpha_s^{(1)}$ and $\alpha_s^{(2)}$ depend on the vessel geometry and its material of construction. By using this expression in (6.12'), we obtain the final form of the energy balances:

$$S_{s,t+1}\Theta_{s,t+1} - S_{st}\Theta_{st} = \delta \sum_{i \in T_s} \sum_{j \in K_i} \left(\frac{f_{ijt}\Theta_{s_{ij},t} + f'_{ijt}\Theta_{s_{ij},t+1}}{2} + \frac{q_{ijt}}{c_p} \right)$$

$$-\delta \sum_{i \in T_s} \sum_{j \in K_i} \frac{f_{ijt}\Theta_{st} + f'_{ijt}\Theta_{s,t+1}}{2}$$

$$-\frac{U_s\delta}{c_p} \frac{(\alpha_s^{(1)} + \alpha_s^{(2)} S_{st})\,(\Theta_{st} - T_t^{amb}) + (\alpha_s^{(1)} + \alpha_s^{(2)} S_{s,t+1})\,(\Theta_{s,t+1} - T_{t+1}^{amb})}{2} \quad \forall\, s, t \qquad (6.12'')$$

6.5.2.5 Capacity and Operability Constraints

The amount of HTM stored in storage tank s cannot exceed the tank capacity C_s. Also, its temperature may have to be maintained within given lower and upper limits $\Theta_s^{\min}$ and $\Theta_s^{\max}$. These restrictions lead to the bounds constraints

$$0 \le S_{st} \le C_s \quad \forall\, s, t \qquad (6.14)$$

and

$$\Theta_s^{\min} \le \Theta_{st} \le \Theta_s^{\max} \quad \forall\, s, t \qquad (6.15)$$

There may also be an upper limit, $F_j^{\max}$, to the flowrate of HTM that can be pumped through any given unit j. We therefore have:

$$0 \le f_{ijt} \le F_j^{\max} \quad \forall\, i, j \in K_i, t \qquad (6.16)$$

6.5.3 Solution of the Scheduling Problem

The scheduling problem that must be solved to determine the optimal operation of the heat integrated plant is summarised below:

Problem [P1]

$$\max \sum_s C_s(S_{s,H+1} - S_{s,0}) - \sum_u \sum_{t=1}^{H} C_{ut} U_{ut} \tag{6.1}$$

subject to

$$S_{s,t+1} = S_{st} + \frac{\delta}{2} \sum_{i\in \overline{T}_s} \sum_{j\in K_i} (f_{ijt} + f'_{ijt}) - \frac{\delta}{2} \sum_{i\in T_s} \sum_{j\in K_i} (f_{ijt} + f'_{ijt}) \quad \forall\, s, t \tag{6.4'}$$

$$q_{ijt} = U_{ij} A_{ij} \left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta} \overline{T}_{ij\theta} - \frac{\Theta_{s_{ij}t} + \Theta_{ijt}^{out}}{2} \right)$$

$$= U_{ij} A_{ij} \left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta} \overline{T}_{ij,\theta+1} - \frac{\Theta_{s_{ij},t+1} + \Theta'^{out}_{ijt}}{2} \right)$$

$$= f_{ijt}\, c_p\, (\Theta_{ijt}^{out} - \Theta_{s_{ij}t})$$

$$= f'_{ijt}\, c_p\, (\Theta'^{out}_{ijt} - \Theta_{s_{ij},t+1}) \quad \forall\, i, j\in K_i, t \tag{6.8}$$

$$q_{ijt} = \sum_{\theta=0}^{p_i-1} (\gamma_{ij\theta}^{(1)} W_{ij,t-\theta} + \gamma_{ij\theta}^{(2)} B_{ij,t-\theta}) \quad \forall\, i, j\in K_i, t \tag{6.10}$$

$$S_{s,t+1}\Theta_{s,t+1} - S_{st}\Theta_{st} = \delta \sum_{i\in \overline{T}_s} \sum_{j\in K_i} \left(\frac{f_{ijt}\Theta_{s_{ij},t} + f'_{ijt}\Theta_{s_{ij},t+1}}{2} + \frac{q_{ijt}}{c_p} \right)$$

$$- \delta \sum_{i\in T_s} \sum_{j\in K_i} \frac{f_{ijt}\Theta_{st} + f'_{ijt}\Theta_{s,t+1}}{2}$$

$$- \frac{U_s\delta}{c_p} \frac{(\alpha_s^{(1)} + \alpha_s^{(2)} S_{st})(\Theta_{st} - T_t^{amb}) + (\alpha_s^{(1)} + \alpha_s^{(2)} S_{s,t+1})(\Theta_{s,t+1} - T_{t+1}^{amb})}{2} \quad \forall\, s, t \tag{6.12''}$$

$$0 \le S_{st} \le C_s \quad \forall\, s, t \tag{6.14}$$

and

$$\Theta_s^{\min} \le \Theta_{st} \le \Theta_s^{\max} \quad \forall\, s, t \tag{6.15}$$

$$0 \le f_{ijt} \le F_j^{\max} \quad \forall\, i, j\in K_i, t \tag{6.16}$$

and all other constraints in the formulation of Kondili *et al.* (1993).

The above mathematical optimisation problem is a nonconvex mixed integer nonlinear programme (MINLP). In particular, the energy balance constraints (6.8) and (6.12'') introduce nonlinearities due to bilinear terms of five different forms, namely $f_{ijt}\Theta^{out}_{ijt}$, $f'_{ijt}\Theta'^{out}_{ijt}$, $S_{st}\Theta_{st}$, $f_{ijt}\Theta_{st}$ and $f'_{ijt}\Theta_{st}$.

In order to circumvent the problems posed by nonlinearity and nonconvexity, we consider an approximate linearisation of the bilinear terms. Specifically, we define the following new variables:

$$\overline{\mathbf{F\Theta}}_{ijt} \equiv f_{ijt}\ \Theta^{out}_{ijt} \qquad \forall\ i,\ j \in K_i,\ t \tag{6.17a}$$

$$\overline{\mathbf{F\Theta}'}_{ijt} \equiv f'_{ijt}\ \Theta'^{out}_{ijt} \qquad \forall\ i,\ j \in K_i,\ t \tag{6.17b}$$

$$\overline{\overline{\mathbf{F\Theta}}}_{ijst} \equiv f_{ijt}\ \Theta_{st} \qquad \forall\ s,\ i \in T_s,\ j \in K_i,\ t \tag{6.17c}$$

$$\overline{\overline{\mathbf{F\Theta}'}}_{ijst} \equiv f'_{ijt}\ \Theta_{s,t+1} \qquad \forall\ s,\ i \in T_s,\ j \in K_i,\ t \tag{6.17d}$$

and

$$\overline{\mathbf{S\Theta}}_{st} \equiv S_{st}\Theta_{st} \qquad \forall\ s,\ t \tag{6.17e}$$

which allows us to rewrite the energy balance equations (6.8) and (6.12′′) in linear form:

$$\begin{aligned}
q_{ijt} &= U_{ij}A_{ij}\left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta}\overline{T}_{ij\theta} - \frac{\Theta_{s_{ij}t} + \Theta^{out}_{ijt}}{2}\right) \\
&= U_{ij}A_{ij}\left(\sum_{\theta=0}^{p_i-1} W_{ij,t-\theta}\overline{T}_{ij,\theta+1} - \frac{\Theta_{s_{ij},t+1} + \Theta'^{out}_{ijt}}{2}\right) \\
&= c_p\,(\overline{\mathbf{F\Theta}}_{ijt} - \overline{\overline{\mathbf{F\Theta}}}_{ijs_{ij}t}) \\
&= c_p\,(\overline{\mathbf{F\Theta}'}_{ijt} - \overline{\overline{\mathbf{F\Theta}'}}_{ijs_{ij}t}) \qquad \forall\ i,\ j \in K_i,\ t
\end{aligned} \tag{6.8′}$$

$$\begin{aligned}
\overline{\mathbf{S\Theta}}_{s,t+1} - \overline{\mathbf{S\Theta}}_{st} = \delta \sum_{i \in \overline{T}_s}\sum_{j \in K_i}\left(\frac{\overline{\overline{\mathbf{F\Theta}}}_{ijs_{ij}t} + \overline{\overline{\mathbf{F\Theta}'}}_{ijs_{ij}t}}{2} + \frac{q_{ijt}}{c_p}\right) \\
- \delta \sum_{i \in T_s}\sum_{j \in K_i} \frac{\overline{\overline{\mathbf{F\Theta}}}_{ijst} + \overline{\overline{\mathbf{F\Theta}'}}_{ijst}}{2}
\end{aligned}$$

$$- \frac{U_s\delta}{c_p}\ \frac{\alpha_s^{(1)}\left(\Theta_{st} + \Theta_{s,t+1} - T^{amb}_t - T^{amb}_{t+1}\right) + \alpha_s^{(2)}\left(\overline{\mathbf{S\Theta}}_{st} + \overline{\mathbf{S\Theta}}_{s,t+1} - S_{st}T^{amb}_t - S_{s,t+1}T^{amb}_{t+1}\right)}{2}$$

$$\forall\, s, t \qquad (6.12''')$$

Lower and upper bounds on the newly introduced variables may be established using appropriate bounds on the variables in the corresponding bilinear terms (McCormick, 1976). This leads to constraints of the form:

$$\max\left(\Theta_{ij}^{out,min} f_{ijt},\ \Theta_{ij}^{out,max} f_{ijt} + f_{ij}^{\max}\Theta_{ijt}^{out} - f_{ij}^{\max}\Theta_{ij}^{out,max}\right) \le \overline{\mathbf{F\Theta}}_{ijt}$$

$$\le \min\left(\Theta_{ij}^{out,max} f_{ijt},\ \Theta_{ij}^{out,min} f_{ijt} + f_{ij}^{\max}\Theta_{ijt}^{out} - f_{ij}^{\max}\Theta_{ij}^{out,min}\right) \quad \forall\, i, j\in K_i, t \qquad (6.18a)$$

$$\max\left(\Theta_{ij}^{\prime out,min} f'_{ijt},\ \Theta_{ij}^{\prime out,max} f'_{ijt} + f_{ij}^{\prime\max}\Theta_{ijt}^{\prime out} - f_{ij}^{\prime\max}\Theta_{ij}^{\prime out,max}\right) \le \overline{\mathbf{F\Theta}}'_{ijt}$$

$$\le \min\left(\Theta_{ij}^{\prime out,max} f'_{ijt},\ \Theta_{ij}^{\prime out,min} f'_{ijt} + f_{ij}^{\prime\max}\Theta_{ijt}^{\prime out} - f_{ij}^{\prime\max}\Theta_{ij}^{\prime out,min}\right) \quad \forall\, i, j\in K_i, t \qquad (6.18b)$$

$$\max\left(\Theta_s^{\min} f_{ijt},\ \Theta_s^{\max} f_{ijt} + f_{ij}^{\max}\Theta_{st} - f_{ij}^{\max}\Theta_s^{\max}\right) \le \overline{\overline{\mathbf{F\Theta}}}_{ijst}$$

$$\le \min\left(\Theta_s^{\max} f_{ijt},\ \Theta_s^{\min} f_{ijt} + f_{ij}^{\max}\Theta_{st} - f_{ij}^{\max}\Theta_s^{\min}\right) \quad \forall\, s, i\in T_s, j\in K_i, t \qquad (6.18c)$$

$$\max\left(\Theta_s^{\min} f'_{ijt},\ \Theta_s^{\max} f'_{ijt} + f_{ij}^{\prime\max}\Theta_{s,t+1} - f_{ij}^{\prime\max}\Theta_s^{\max}\right) \le \overline{\overline{\mathbf{F\Theta}}}'_{ijst}$$

$$\le \min\left(\Theta_s^{\max} f'_{ijt},\ \Theta_s^{\min} f'_{ijt} + f_{ij}^{\prime\max}\Theta_{s,t+1} - f_{ij}^{\prime\max}\Theta_s^{\min}\right) \quad \forall\, s, i\in T_s, j\in K_i, t \qquad (6.18d)$$

$$\max\left(\Theta_s^{\min} S_{st},\ \Theta_s^{\max} S_{st} + C_s\Theta_{st} - C_s\Theta_s^{\max}\right) \le \overline{\mathbf{S\Theta}}_{st}$$

$$\le \min\left(\Theta_s^{\max} S_{st},\ \Theta_s^{\min} S_{st} + C_s\Theta_{st} - C_s\Theta_s^{\min}\right) \quad \forall\, s, t \qquad (6.18e)$$

The lower and upper bounds $\Theta_s^{\min}$ and $\Theta_s^{\max}$ on the temperatures Θ_{st}, and the upper bound C_s on the HTM holdups S_{st} are given as part of the problem specification. However, appropriate upper bounds $f_{ij}^{\max}$ and $f_{ij}^{\prime\max}$ for the flowrates f_{ijt} and f'_{ijt} at the start and end of each time interval must be established. Similarly, we need lower bounds $\Theta_{ij}^{out,min}$ and $\Theta_{ij}^{\prime out,min}$ and upper bounds $\Theta_{ij}^{out,max}$ and $\Theta_{ij}^{\prime out,max}$ on the HTM exit temperatures Θ_{ijt}^{out} and $\Theta_{ijt}^{\prime out}$ at the start and end of the time intervals. The derivation of these bounds from the energy balance equations (6.5) and (6.6) and pumping rate constraints (6.16) is presented in the Appendix C.

The modified linear energy balance equations (6.8′) and (6.12‴) together with the linear bounds (6.18) can be used to replace the nonlinear energy balances (6.8) and (6.12′) in the scheduling formulation of problem [P1]. Hereafter we refer to the MILP resulting from this substitution as **Problem [P2]**.

Because of its linear nature, problem [P2] can, at least in principle, always be solved using an appropriate method such as branch-and-bound. On the other hand, it should be noted that [P2] is only a relaxation of [P1] and therefore the temperature, flowrate and holdup profiles obtained by solving it may not satisfy the original energy balance equations. Also the objective function at the optimal solution of the MILP will be only an upper bound on the optimal objective function that can be attained by the system.

In view of the above discussion, we must consider modifications to the standard branch-and-bound algorithm to be applied to this problem. In particular, we note that each integer-feasible node visited during the search for the solution of [P2] corresponds to a feasible allocation of the processing equipment as determined by the corresponding values of the binary variables W_{ijt}. However, due to the relaxation of the bilinearities, the corresponding value of the objective function is generally an upper bound to the true value of the objective function at this node.

Now consider solving the restricted nonlinear programming (NLP) problem obtained from [P1] by fixing the binary W_{ijt} variables at the values that they have at an integer-feasible node. This corresponds to the solution of a non-convex NLP, a feasible solution to which is also a feasible solution to [P1]. However, because of the non-convexity, standard NLP algorithms are guaranteed to obtain only a local solution. This therefore provides a *lower* bound on the true objective function value for this node.

In summary, at each integer-feasible node of the branch-and-bound search for the solution of [P2], we establish two bounds on the corresponding objective function value of [P1]: an upper one from [P2], and a lower one from the (restricted) [P1].

If the search were to examine *all* integer-feasible solutions of [P2], then the maximum lower bound and the maximum upper bound encountered during this search would also bound the optimal solution of [P1]. However, if at any point during the search, the upper bound to the objective function at the node being examined is smaller than the best lower bound established so far, then it is clearly not necessary to consider further this node or any of its descendants.

In fact, practical implementations of branch-and-bound algorithms for solving MILP problems usually determine the solution only to within a margin of optimality $\lambda \in [0, 1)$ such that

$$\texttt{Objective Function obtained} \geq (1-\lambda) \times \texttt{True Objective Function}$$

Typically λ is in the range 0.01 to 0.1. In the context of the modified branch-and-bound algorithm discussed here, a node will be considered to be fathomed if the corresponding objective function of [P2] (*i.e.* the upper bound on the objective function of [P1]) does not exceed

$$\frac{\phi^{LB}}{1-\lambda}$$

where ϕ^{LB} is the objective function of the best feasible solution to [P1] (*i.e.* the best lower bound to the optimal solution of [P1]) obtained so far in the search.

The modified branch-and-bound algorithm is stated formally in figure 6.14.

Given a margin of optimality $\lambda \in [0, 1)$,

1 Set lower and upper bounds on the objective function, $\phi^{LB} := -\infty$, $\phi^{UB} := -\infty$.

2 Solve linear programme obtained from [P2] by relaxing the integrality requirements on the equipment allocation variables W_{ijt}.

3 If feasible, place the solution obtained on a stack containing unexplored search tree nodes.

4 WHILE the stack is not empty DO

4.1 Select and remove a solution from the stack. Denote its objective function value by ϕ^{node}.

4.2 IF $(1 - \lambda)\,\phi^{node} > \phi^{LB}$ THEN

4.2.1 IF an integer-feasible node, THEN

4.2.1.1 Set $\phi^{UB} := \max(\phi^{UB}, \phi^{node})$.

4.2.1.2 Solve NLP problem derived from [P1] by fixing the W_{ijt} at their current integer values. If the problem is found to be feasible, then denote the objective function value by ϕ^{NLP}, otherwise set $\phi^{NLP} := -\infty$.

4.2.1.3 IF $\phi^{NLP} \geq \phi^{LB}$ THEN

4.2.1.3.1 Set $\phi^{LB} := \phi^{NLP}$.

4.2.1.3.2 Record values of all variables as best solution currently available.

4.2.1.4 IF $\phi^{NLP} < \phi^{node}$ THEN

4.2.1.4.1 Add integer cut to exclude current combination of W_{ijt}. Solve LP; if feasible, place solution on stack.

4.2.2 ELSE

4.2.2.1 Select a W_{ijt} with a non-integral value.

4.2.2.2 Fix $W_{ijt} := 0$ and solve the corresponding linear programme; if feasible, place solution on stack.

4.2.2.3 Fix $W_{ijt} := 1$ and solve the corresponding linear programme; if feasible, place solution on stack.

End IF

4.3 ELSE

4.3.1 Set $\phi^{UB} := \max(\phi^{UB}, \phi^{node})$.

End IF

End WHILE

Figure 6.14: Modified Branch-and-bound Algorithm

Note that with a non-zero margin of optimality λ, an integer-feasible node may be fathomed even if its objective function value actually exceeds (slightly) the best lower bound ϕ^{LB}

found so far. In this case, we save the solution of the NLP at step 4.2.1.2, which could indeed have resulted in an improvement in ϕ^{LB} at step 4.2.1.3.1. On the other hand, irrespective of whether the fathomed node was integer-feasible or not, ϕ^{node} is a valid upper bound on the optimal objective function of [P1], and therefore step 4.3.1 is needed to ensure that this bound is updated. However, we note that this step never results in a change in the value of the upper bound ϕ^{UB} if λ is set to zero.

It should be noted that usually the integer feasible solution examined at step 4.2.1 results from only a *partial* assignment of the binary variables W_{ijt}, with all other W_{ijt} variables simply happening to attain binary values in the solution of the LP relaxations (steps 4.2.2.2 and 4.2.2.3). However the NLP solved at step 4.2.1.2 is derived from [P1] by fixing *all* W_{ijt} to their current values. It is therefore possible that an NLP corresponding to a different all-integer descendant of this node will actually have a better objective function value than the one obtained at step 4.2.1.2. This eventuality is catered for by step 4.2.1.4: the node is considered to be fathomed only if the solution of the current NLP is the best possible (*i.e.* $\phi^{NLP} = \phi^{node}$). Otherwise an integer cut excluding the current combination of W_{ijt} variables is added to the formulation, and the corresponding linear relaxation is solved. The node is then placed back on the stack so as to ensure that its integer feasible descendants are considered later in the search.

Upon termination of the algorithm, we obtain a range within which the optimal objective function of [P1], ϕ^* is guaranteed to lie:

$$\phi^{LB} \leq \phi^* \leq \phi^{UB} \tag{6.19}$$

A feasible solution of [P1] corresponding to ϕ^{LB} is also obtained provided that $\phi^{LB} > -\infty$. This will always be the case if at least one of the NLPs solved at step 4.2.1.2. manages to determine a feasible local minimum solution.

Failures may occur at step 4.2.1.2 either because the NLP is indeed infeasible for the given values of the binary variables, or because the NLP code simply cannot obtain a solution to the problem posed to it. In practice, the probability of the latter mode of failure in the NLP code and the computational cost of solving the NLP are both drastically reduced by initialising the NLP iterations at the values of the variables obtained by solving the linear programming relaxation for this node (steps 4.2.2.2 and 4.2.2.3).

The algorithm presented here bears some similarities to the LP/NLP algorithm of Quesada and Grossmann (1992). Both algorithms construct MILP approximations of MINLP problems, which they proceed to solve using branch-and-bound type search procedures. Both also solve the restricted NLPs at each integer-feasible node encountered during the search.

The main difference is that the Quesada and Grossmann algorithm is restricted to the solution of *convex* MINLPs and can therefore derive the approximating MILP by replacing the nonlinear functions in the original problem by Taylor expansions at a suitable initial point. In our case, because of the nonconvexity of the problem, we use the alternative McCormick (1976) linearisations of the bilinear terms. This also allows us to deal with nonlinear equality constraints directly without the need of eliminating them from the problem.

Another difference is that the solutions of the NLPs at integer-feasible nodes in the Quesada and Grossmann algorithm are used to generate further linearisations that are subsequently appended to the original MILP in an attempt to provide a better characterisation of the feasible region. In our case, a similar extension of the MILP is neither easy nor, in our experience, necessary: for this particular type of problem, the original linear approximation to the nonlinear constraints has been found usually to be fairly accurate (see results presented in next section).

An alternative to using a local optimisation method for the solution of the NLP at each integer-feasible node (step 4.2.1.2) would be to employ a global optimisation technique such as those based on spatial branch-and-bound methods. In this case, the final value of ϕ^{LB} would also be the optimal solution of [P1] (provided that λ were set to zero), and there would be no need for calculating an upper bound ϕ^{UB}. On the other hand, if ϕ^{LB} and ϕ^{UB} are close, using a global optimisation algorithm at step 4.2.1.2 is unnecessary.

6.5.4 Example of Indirect Heat Integration

We consider the same example process as for the case of direct heat exchange, but now we assume that heat is exchanged via a transfer medium which can be kept in tanks as shown in figure 6.15.

Details of the properties of the HTM and the heat-integrated tasks are shown in tables 6.1 and 6.2 respectively.

Storage Tank	Capacity (te)	Initial Conditions		Operating Temperature Limits	
		Amount (te)	Temperature (K)	Minimum (K)	Maximum (K)
HOT	100	0	398	393	405
COLD	100	100	373	363	378
HTM specific heat capacity: 2.1 $MJ/te.K$					

Table 6.1: Properties of HTM States

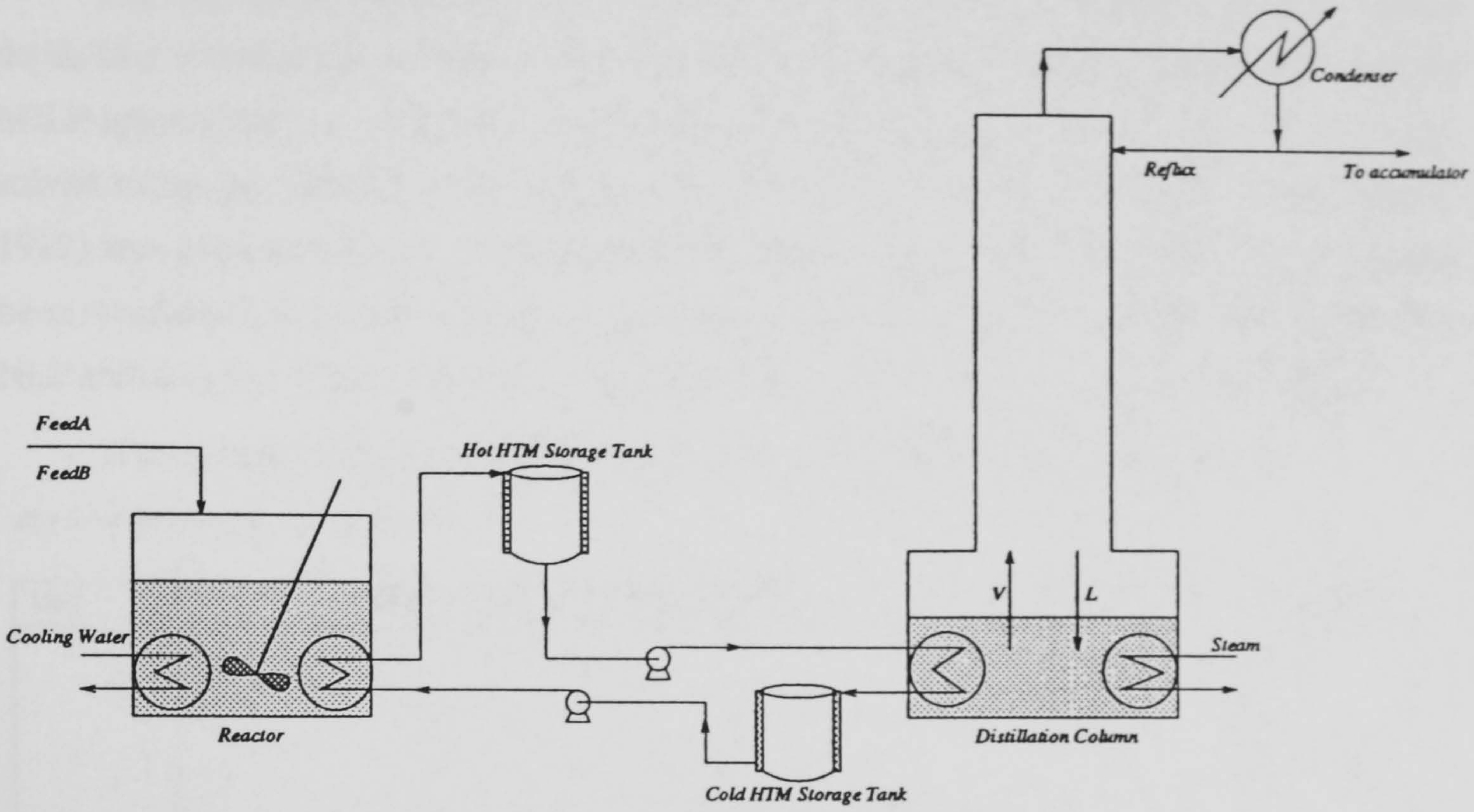


Figure 6.15: Indirect Heat Exchange Network for Example Process

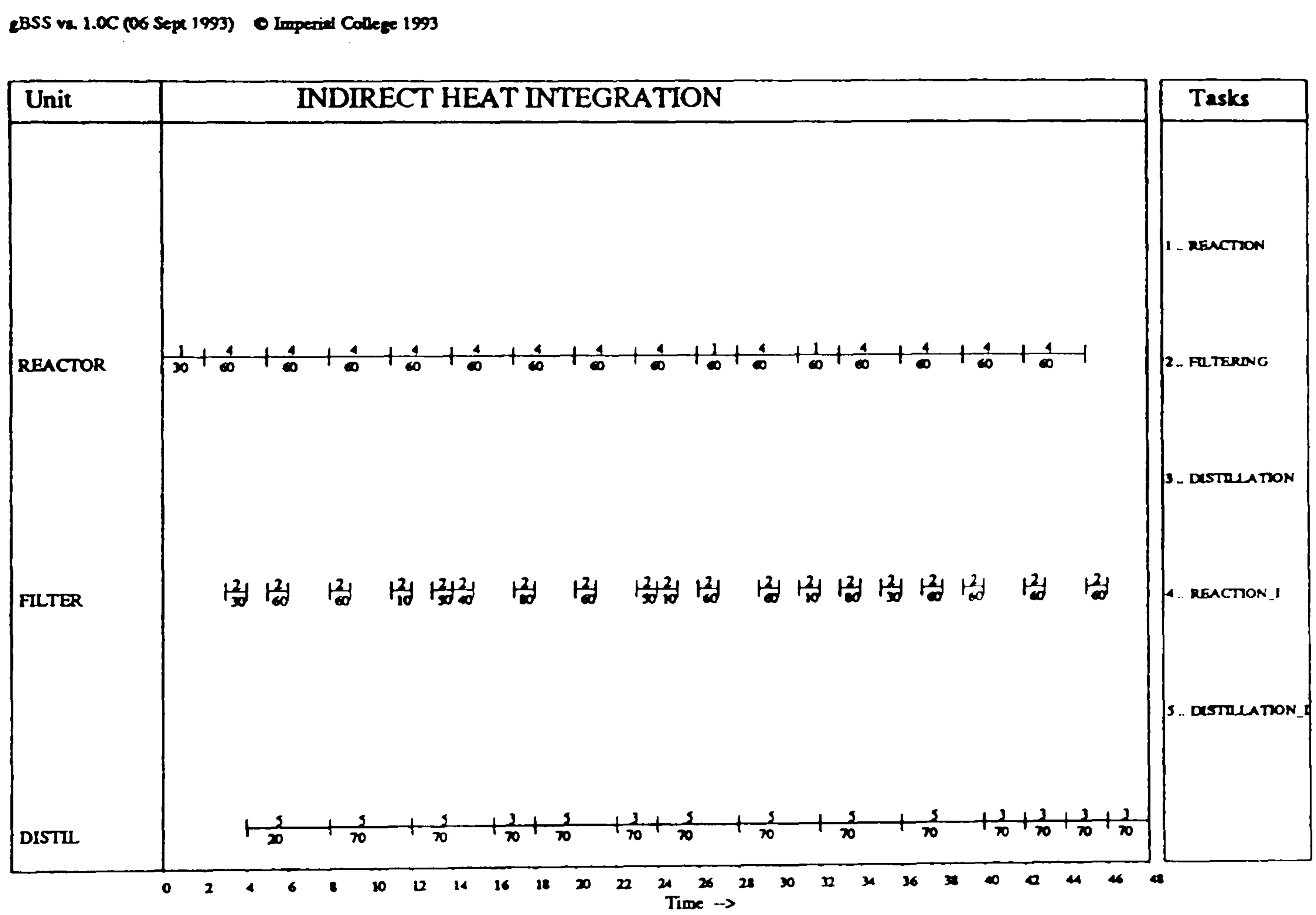
Heat-Integrated Task	Duration p_i (hr)	Heat Transfer Characteristics $U_{ij}A_{ij}$ $(MJ/hr.K)$	Max. HTM Flowrate F_j^{max} (te/hr)	Time θ (hr)	Operating Temperature $\bar{T}_{ij\theta}$ (K)	HTM Heat Load $\gamma_{ij\theta}^{(1)}$ (MJ/hr)	HTM Heat Load $\gamma_{ij\theta}^{(2)}$ $(MJ/hr.te)$
Reaction	3.0	17.0	25.0	0	405		
						0.0	0.0
				1	420		
						100.0	7.0
				2	415		
						100.0	6.0
				3	410		
Distillation	4.0	11.0	25.0	0	350		
						-50.0	-4.0
				1	352		
						-50.0	-4.0
				2	354		
						-50.0	-4.0
				3	356		
						-50.0	-4.0
				4	358		

Table 6.2: Properties of Heat-integrated Tasks

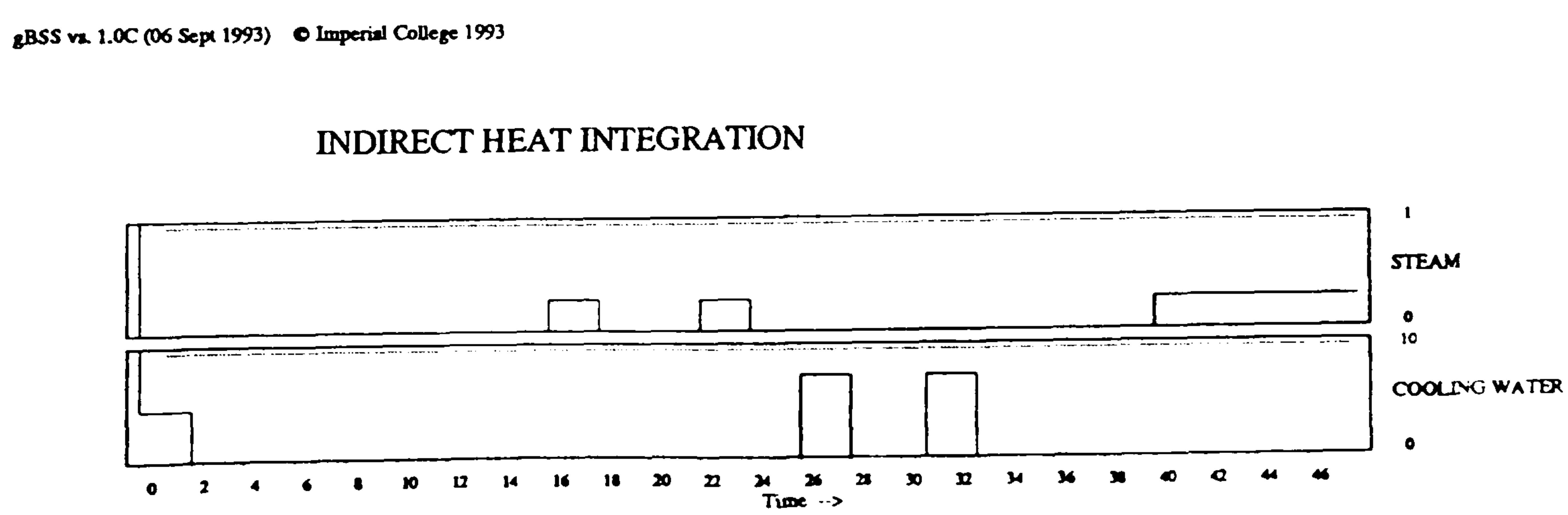
The overall heat transfer coefficient to the environment is taken to be 0.8 $W/m^2.K$ (= 0.00288 $MJ/hr.m^2.K$). The effective heat transfer area is taken to be directly proportional to the amount of material held in the vessel, with $\alpha_s^{(1)} = 0.0\ m^2$ and $\alpha_s^{(2)} = 1.1\ m^2/te$. Finally, the ambient temperature is taken to be constant at 283 K throughout the time horizon.

The algorithm presented in the previous section was implemented using a simple depth-first search implementation of the branch-and-bound algorithm in order to solve the MILP approximation. The linear programming relaxations (steps 2, 4.2.2.2 and 4.2.2.3) were solved using the MINOS code (Murtagh and Saunders, 1983). The CONOPT code (Drud, 1992) was used to solve the restricted NLPs at step 4.2.1.2 of the algorithm. As has already been mentioned, the solution of the LP at each integer-feasible node was used to initialise the NLP solution procedure. A margin of optimality (λ) of 0.05 was used for the calculation.

The optimal schedule and utility utilisation profiles are shown in figure 6.16.



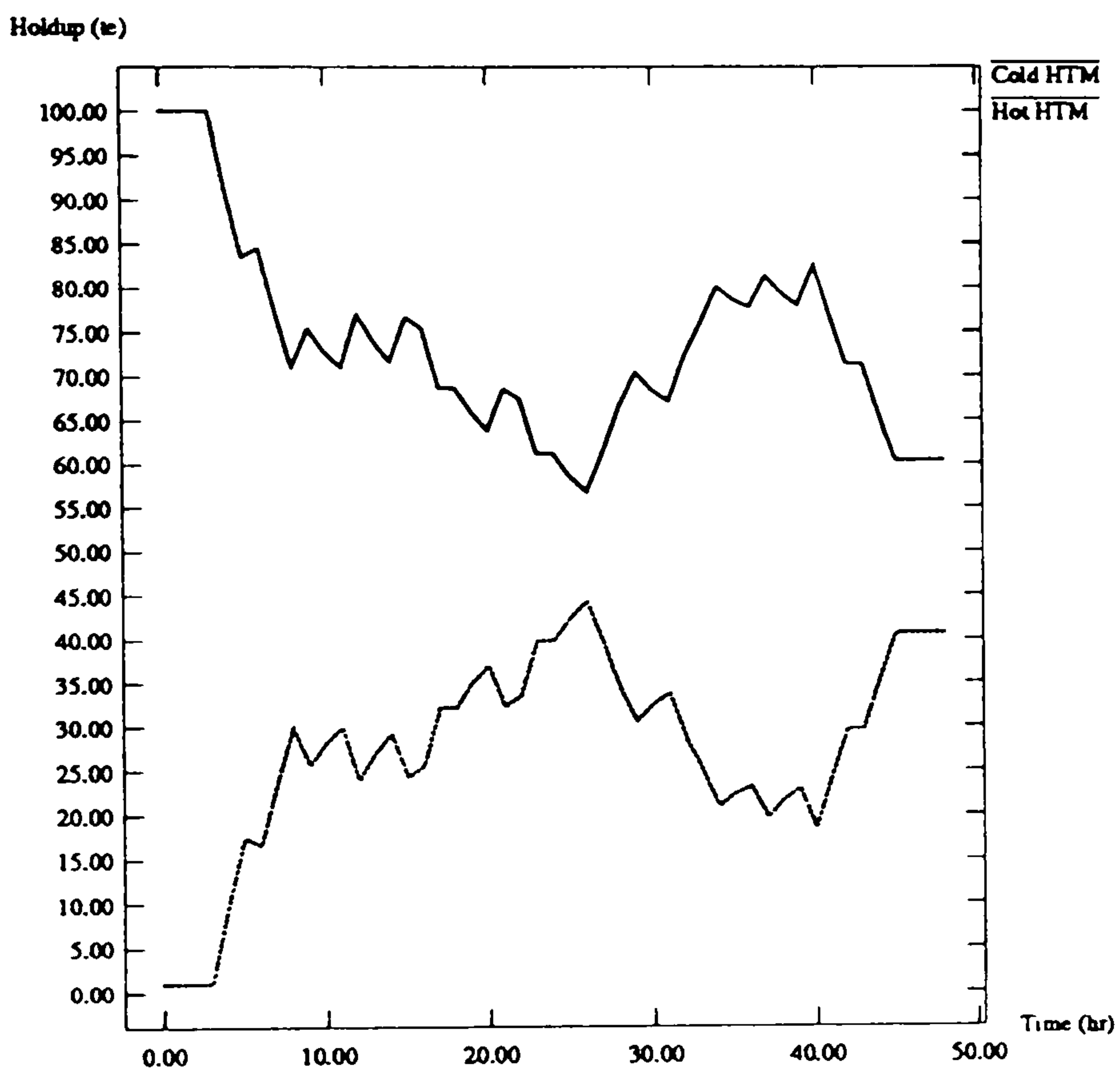
(a) Utilisation of Processing Equipment



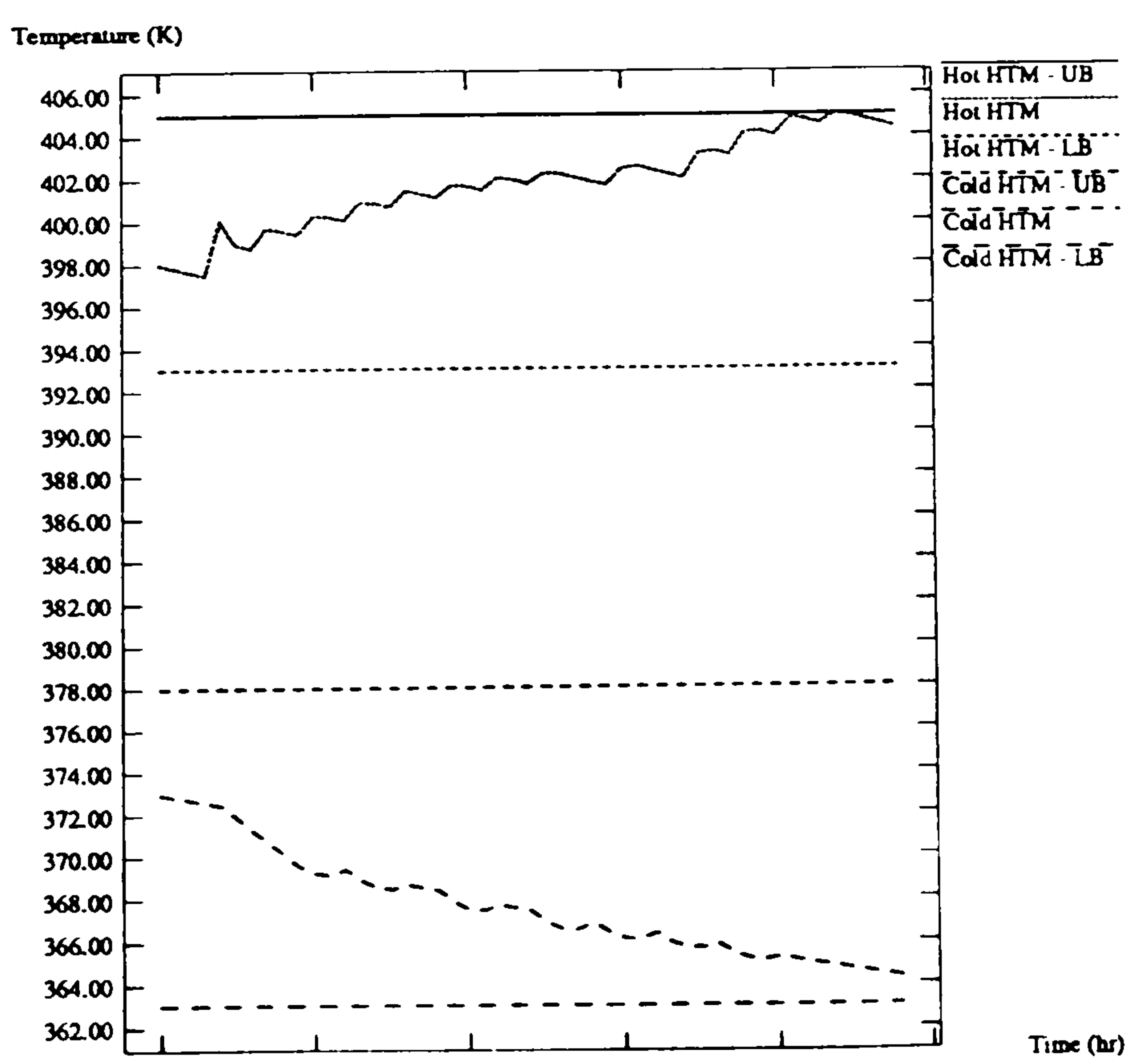
(b) Utilisation of External Utilities

Figure 6.16: Optimal Schedule with Indirect Heat Integration

The amounts of the two products produced are 698.1 and 232.7 *te* respectively, while the costs of the external steam and cooling water utilisation are 693.7 and 158.9 *rcu* respectively. The corresponding value of the objective function is 3801.4 *rcu*.



(a) HTM Holdups



(b) HTM Temperatures

Figure 6.17: Holdup and Temperature Profiles for HTM Storage Tanks

Figure 6.17 shows the time profiles for the holdups and the temperatures in the two HTM storage tanks.

It can be seen that these variables are held within their specified bounds. Note that the temperature of the hot HTM reaches its upper bound near the end of the time horizon. This is mainly due to the fact that the last few reaction batches use HTM as a cooling medium, while the corresponding distillation batches are carried out on a stand-alone basis (*cf.* figure 6.16a).

The flowrates of the HTM through the two heat-integrated operations are shown in figure 6.18, which clearly demonstrates the discontinuous nature of these quantities.

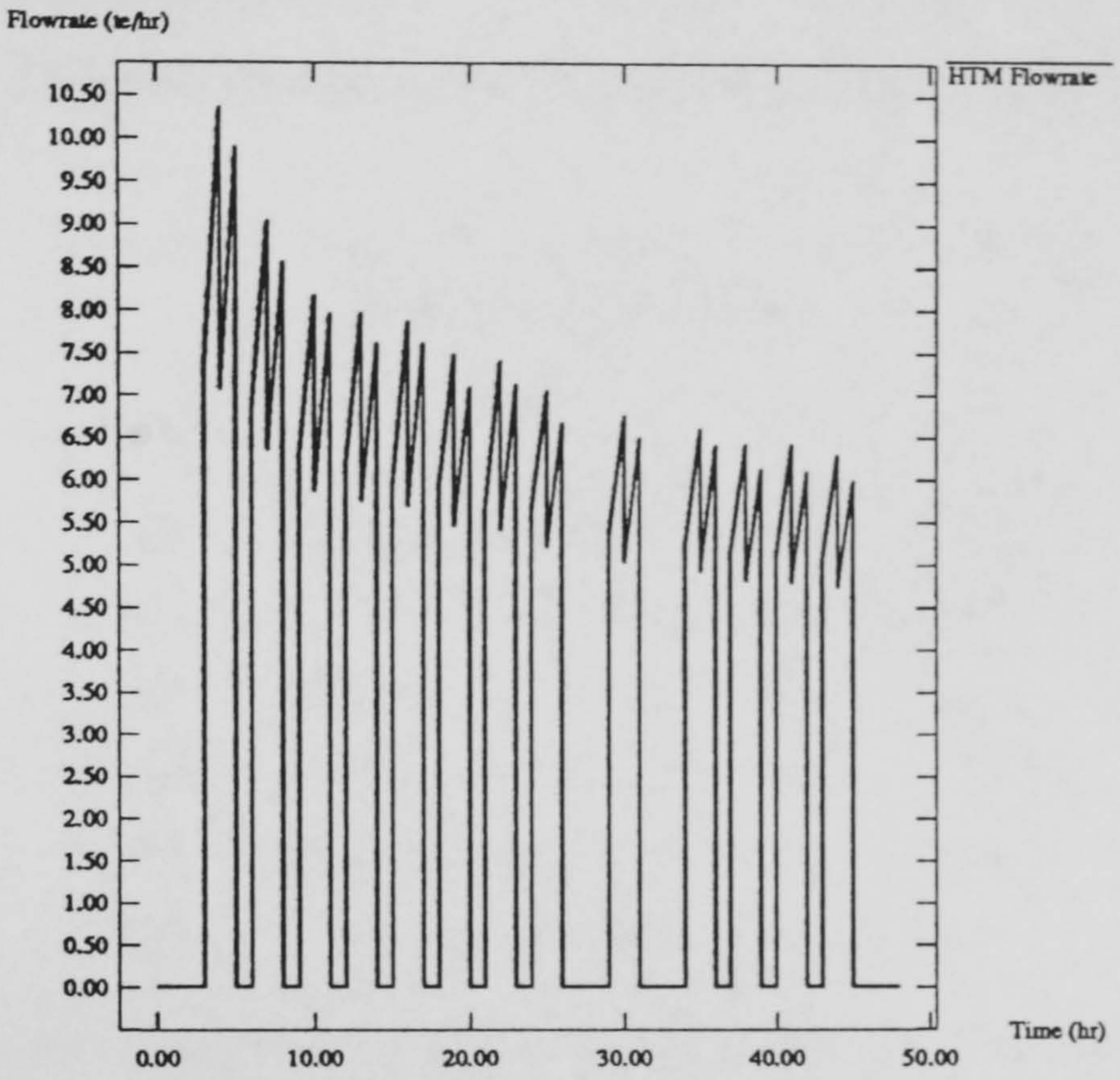
The small HTM flowrate through the first batch of the heat-integrated distillation task in figure 6.18b is a consequence of the smaller than usual size of this batch (*cf.* figure 6.16a).

Problem size statistics for this problem are shown in table 6.3.

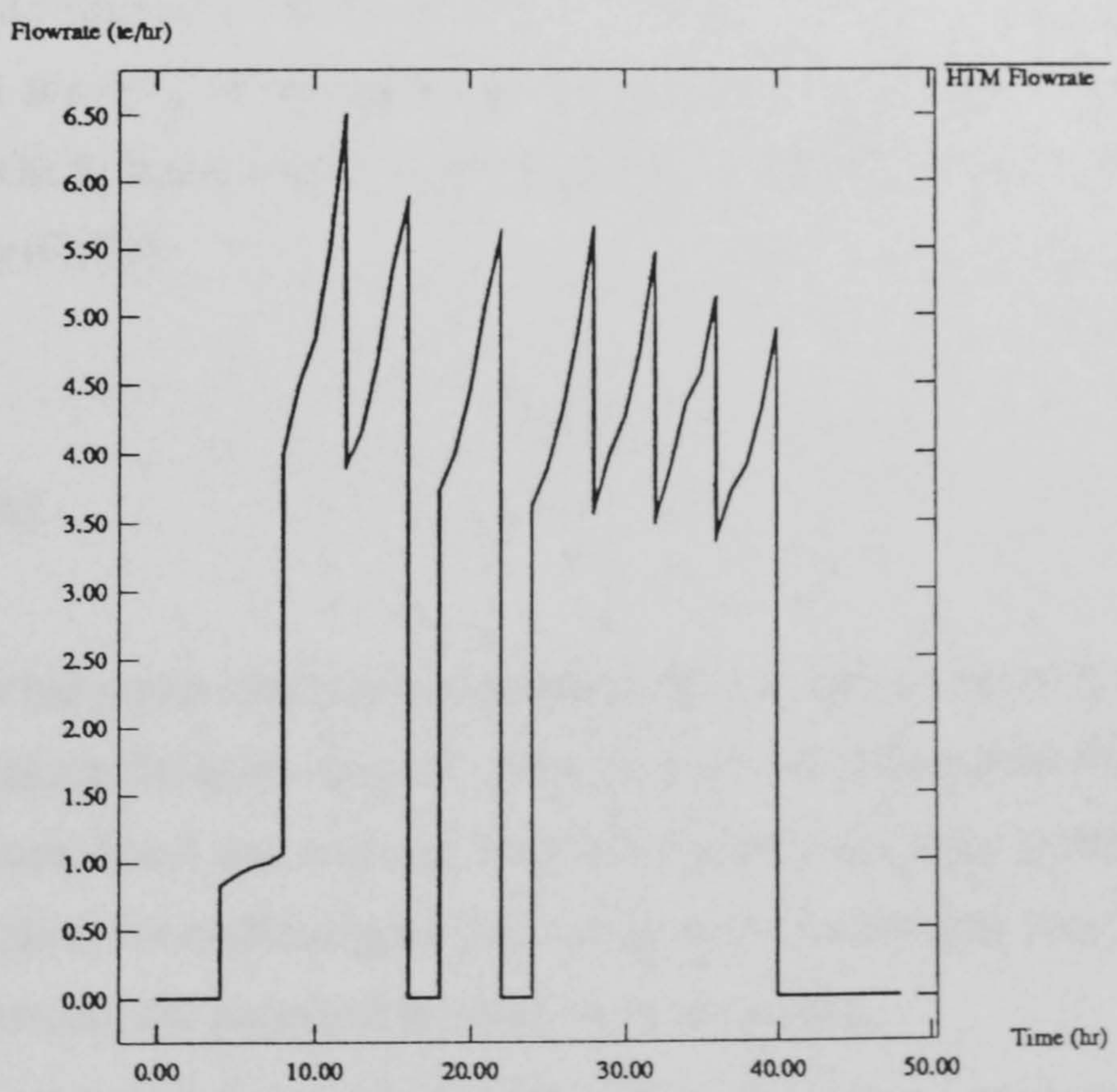
MINLP [P1]		
Binary variables	:	233
Continuous variables	:	1041
Constraints	:	1548
MILP [P2]		
Binary variables	:	233
Continuous variables	:	1781
Constraints	:	3601
NLP (step 4.2.1.2)		
Variables	:	1041
Constraints	:	938

Table 6.3: Problem Size Statistics for Indirect Heat Integration Example

It can be seen that the MILP problem [P2] is significantly larger than the original MINLP problem [P1] due to the introduction of the new variables (17) representing bilinear products, and the bounding constraints (18). On the other hand, the NLP problem solved at step 4.2.1.2 of the algorithm has fewer constraints than the MINLP because, for a fixed set of binary variables, the equipment allocation constraints can be omitted from the formulation, and the equipment capacity constraints become simple bounds on the batchsize variables.



(a) HTM flowrate through reactor



(b) HTM Flowrate through Distillation Column

Figure 6.18: HTM Flowrates through Heat-integrated Operations

The progress of the branch-and-bound search is shown in Table IV in terms of the integer-feasible nodes encountered.

Objective function for fully relaxed LP (step 2): 4030.6

Integer Feasible Nodes				
Node	ϕ^{node}	ϕ^{NLP}	ϕ^{LB}	ϕ^{UB}
88	3589.0	3589.0	3589.0	3589.0
116	3783.0	3742.2	3742.2	3783.0
117	3783.0	3752.6	3752.6	3783.0
118	3783.0	3752.6	3752.6	3783.0
219	3924.2	3801.4	3801.4	3924.2
230	Termination			

Table 6.4: Search Summary for Indirect Heat Integration Example

It can be seen that the differences between the solution of the LP relaxation and the NLP are generally very small, and very often zero. Also the final values of the upper and lower bounds bracket the optimal solution to within less than 3.2%. Furthermore, the linear relaxation of the MILP [P2] solved at step 2 of the algorithm is reasonably tight, the corresponding objective function value (4030.6) being within 6.0% of the lower bound (3801.4) on the optimal objective function value of [P1].

6.6 Conclusions

This chapter has considered the exploitation of heat integration in the operation of multipurpose plants, taking detailed account of its interactions with production scheduling. The incorporation of both direct and indirect heat integration within the framework of a general scheduling formulation for multipurpose plant production scheduling was examined, and simple examples illustrating the potential benefits were presented.

Direct heat integration involved only relatively minor extensions to the general scheduling formulation of Kondili *et al.* (1993). On the other hand, dealing with the complexities of indirect heat integration necessitated a detailed characterisation of the mass and energy holdups of the HTM used for effecting the integration. In addition to requiring a number of new constraints, this extension changed the mathematical nature of the underlying scheduling formulation through the introduction of both nonlinearities and nonconvexities.

A modified branch-and-bound procedure for solving the resulting nonconvex MINLP has been proposed. This involves the solution of nonconvex NLPs, and therefore in principle it cannot guarantee always to obtain the global optimum, or indeed even a feasible solution to the problem. However, the solution procedure generates upper and lower bounds on the global optimum, and in our experience these tend to define a fairly narrow range within which the true solution must lie.

Although direct and indirect energy integration were examined separately in this chapter, they can clearly co-exist within the same plant. Also, a given plant may involve more than one independent heat transfer network for indirect heat integration. These generalisations can readily be accommodated with the concepts presented here.

Furthermore, although this chapter concentrated on the introduction of heat integration considerations to a short-term scheduling formulation, the approach presented is equally applicable to *any* one of the family of formulations (*e.g.* periodic scheduling) based on the STN framework (see Pantelides, 1994).

One fundamental assumption in this chapter is that the detailed operating characteristics of heat-integrated operations are fixed and known *a priori*. This raises important questions regarding the optimal manner in which two (or more) heat-integrated processing steps are to be operated, and the interaction of operating decisions made at this level with decisions at the plant scheduling level. The use of dynamic optimisation for the determination of optimal policies for pairs of heat-integrated batch unit operations is described in the next chapter.

Chapter 7

Optimal Operation of Thermally Coupled Batch Operations

The problem of establishing optimal operating policies for a pair of thermally coupled batch operations is studied in this chapter. This involves the determination of optimal time-varying profiles for control variables relating to each of the two operations individually, as well as to the heat exchange between them. The durations of the two operations, and the optimal offset between their starting times also need to be established.

Detailed dynamic models based on mixed sets of differential and algebraic equations are used to describe the transient behaviour of the processes under consideration. The optimisation is formulated as a multistage optimal control problem with the objective of maximising the overall rate of value-added, and an algorithm based on control vector parameterisation is employed for its solution.

The application of the technique is illustrated by two examples involving an exothermic reaction and a distillation step, with and without material coupling.

7.1 Basis of Present Work

Batch unit operations are often significant consumers or producers of energy and other utilities. However, the problem of energy integration in batch plants is complicated by the time-varying nature of the operations involved, the interactions between energy integration and other control variables that can be used to manipulate the individual operations, and the influence that energy integration may have on the overall plant scheduling problem.

One common assumption underlying the work on energy integration (see chapter 6) is that the processing steps being integrated are simple cooling and heating operations that can be described fully by rather simple models. However, more complex processing steps (such as exothermic or endothermic reactions, and batch distillation) involve strong interactions between the consumption of utilities and the overall performance of the operation itself measured in terms of the attainment of targets relating to the quantity and quality of the product, as well as the total processing time. The latter is an important factor affecting both the attainable throughput of the individual operation and the plant scheduling.

Moreover, it must be recognised that, in many cases, the rate of utility provision is often only one of several possible controls that can be applied. For instance, in a batch distillation step, the reflux ratio is a commonly used control in addition to the heating and cooling loads to the reboiler and condenser respectively.

The possibility of using energy integration to replace or supplement the provision of external utilities introduces further complexity in the problem. This is true even in the simplest case of two heat-integrated batch operations. In addition to the time profile of the rate of heat exchange (*e.g.* as characterised by the flow of a heat transfer medium) between the two operations, the durations of the latter and an appropriate offset between their starting times must be established.

The above considerations lead to a complex time-varying optimisation problem which, in general, can only be solved through the use of fairly detailed dynamic models describing the transient behaviour of the individual operations and the heat exchange network between them.

This chapter addresses the problem of determining optimal operating policies for pairs of thermally-coupled batch operations. Some complications arising from the potential existence of *material* coupling between the two operations (*i.e.* the product of the first operation forming the feed of the second) are also considered.

7.2 Mathematical Formulation

We consider the heat integration of two batch operations, the behaviour of which is described by mixed sets of differential and algebraic equations (DAEs) of the form:

$$f^{(k)}\left(x^{(k)},\, \dot{x}^{(k)},\, y^{(k)},\, u^{(k)},\, \tilde{u}^{(k)},\, v^{(k)}\right) = 0; \quad t \in [0, \tau^{(k)}], \quad k = 1, 2 \tag{7.1}$$

where $x^{(k)}(t)$ and $y^{(k)}(t)$ are the sets of differential and algebraic variables respectively for the kth operation. The duration of the batch operation is denoted by $\tau^{(k)}$, while time t is measured relatively to the start of each individual operation.

Both $u^{(k)}(t)$ and $\tilde{u}^{(k)}(t)$ are sets of control variables; $\tilde{u}^{(k)}(t)$ are variables determining the rate of heat exchange taking place at time t, whereas $u^{(k)}(t)$ comprise all other controls. For instance, in the case of a batch distillation operation, the reboiler of which is thermally coupled with an exothermic reactor through an intermediate heat transfer medium, the flowrate of the latter would belong to $\tilde{u}^{(k)}$ whilst $u^{(k)}$ could include the reflux ratio and the flowrates of external utilities (steam and cooling water). Finally, $v^{(k)}$ denotes a set of time-invariant parameters that may affect the batch operation.

The initial condition of operation k can, in general, be specified by a set of nonlinear algebraic relations between the initial values of the variables, *i.e.*

$$I^{(k)}\left(x^{(k)}(0),\ \dot{x}^{(k)}(0),\ y^{(k)}(0),\ u^{(k)}(0),\ \bar{u}^{(k)}(0),\ v^{(k)}\right)=0; \quad k=1,2 \tag{7.2}$$

Of course, in most cases, the above takes the much simpler form of specifying values for a subset of the variables $x^{(k)}(0)$, $\dot{x}^{(k)}(0)$ and $y^{(k)}(0)$. This specification may be in terms of either constant numerical values or some of the time-invariant parameters $v^{(k)}$. For instance, the initial batch size to be charged into a reactor may be treated as a parameter $v^{(k)}$ to be determined by the optimisation.

The duration of the batch operation $\tau^{(k)}$ may be specified either explicitly or implicitly, in terms of one or more end-point equality or inequality constraints relating, for instance, to the quality and quantity of the product. These are of the form:

$$c_{EQ}^{(k)}\left(x^{(k)}(\tau^{(k)}),\ \dot{x}^{(k)}(\tau^{(k)}),\ y^{(k)}(\tau^{(k)}),\ u^{(k)}(\tau^{(k)}),\ \bar{u}^{(k)}(\tau^{(k)}),\ v^{(k)}\right)=0; \quad k=1,2 \tag{7.3a}$$

and

$$c_{IN}^{(k)}\left(x^{(k)}(\tau^{(k)}),\ \dot{x}^{(k)}(\tau^{(k)}),\ y^{(k)}(\tau^{(k)}),\ u^{(k)}(\tau^{(k)}),\ \bar{u}^{(k)}(\tau^{(k)}),\ v^{(k)}\right)\le 0; \quad k=1,2 \tag{7.3b}$$

Furthermore, the batch operation may be subject to inequality path constraints that may reflect safety or product quality considerations. These are of the general form:

$$r^{(k)}\left(x^{(k)}(t),\ \dot{x}^{(k)}(t),\ y^{(k)}(t),\ u^{(k)}(t),\ \bar{u}^{(k)}(t),\ v^{(k)}\right)\le 0; \quad t\in[0,\tau^{(k)}], \quad k=1,2 \tag{7.4}$$

The control variables $u^{(k)}$ and $\bar{u}^{(k)}$, and the time invariant parameters $v^{(k)}$ may also be subject to given lower and upper bounds:

$$u_L^{(k)}\le u^{(k)}(t)\le u_U^{(k)}; \quad \bar{u}_L^{(k)}\le \bar{u}^{(k)}(t)\le \bar{u}_U^{(k)}; \quad v_L^{(k)}\le v^{(k)}\le v_U^{(k)}; \quad k=1,2 \tag{7.5}$$

Finally, the nett value added by a single batch of material undergoing a given operation can normally be expressed in terms of the final values of a subset of the describing variables:

$$\phi^{(k)}\left(x^{(k)}(\tau^{(k)}),\ \dot{x}^{(k)}(\tau^{(k)}),\ y^{(k)}(\tau^{(k)}),\ u^{(k)}(\tau^{(k)}),\ \bar{u}^{(k)}(\tau^{(k)}),\ v^{(k)}\right); \quad k=1,2 \tag{7.6}$$

This may include the value of the products, as well as the cost of the raw materials, and the cumulative cost of any external utilities used over the duration of the operation.

We now turn our attention to the optimisation problem. We start by considering the two operations in the absence of *any* thermal or material coupling. In such cases, the control variables $\bar{u}^{(k)}$ describing the heat integration are fixed to some nominal value $\bar{u}_*^{(k)}$ (*e.g.* zero flowrate for the heat transfer medium), and the overall objective is the maximisation of the rate of value added expressed as the value of each batch divided by the time required for its production. The latter is given by the sum of the task duration $\tau^{(k)}$ and any set-up time $\bar{\tau}^{(k)}$ required for filling the corresponding vessel with the raw materials, emptying the final contents, cleaning *etc*. This leads to the optimisation problem:

$$\max_{u^{(k)}(t),\, v^{(k)},\, \tau^{(k)};\; k=1,2} \sum_{k=1}^{2} \frac{\phi^{(k)}\left(x^{(k)}(\tau^{(k)}),\, \dot{x}^{(k)}(\tau^{(k)}),\, y^{(k)}(\tau^{(k)}),\, u^{(k)}(\tau^{(k)}),\, \bar{u}_*^{(k)},\, v^{(k)}\right)}{\tau^{(k)} + \bar{\tau}^{(k)}} \tag{7.7}$$

subject to constraints (7.1)-(7.5). Of course, in reality, this problem decouples into two separate optimisation problems, one for each of the two operations.

If we now introduce the possibility of thermal coupling, the two operations will operate with an offset θ between their respective starting times, where θ is also to be determined by the optimisation. For direct energy integration, the two operations must overlap in time at least partially, in the manner shown in figure 7.1. During the period of overlap, the control variables $\bar{u}^{(k)}(t)$ are related through:

$$\bar{u}^{(1)}\,(t+\theta) = \bar{u}^{(2)}\,(t) \tag{7.8}$$

while, outside it, they revert to their nominal values.

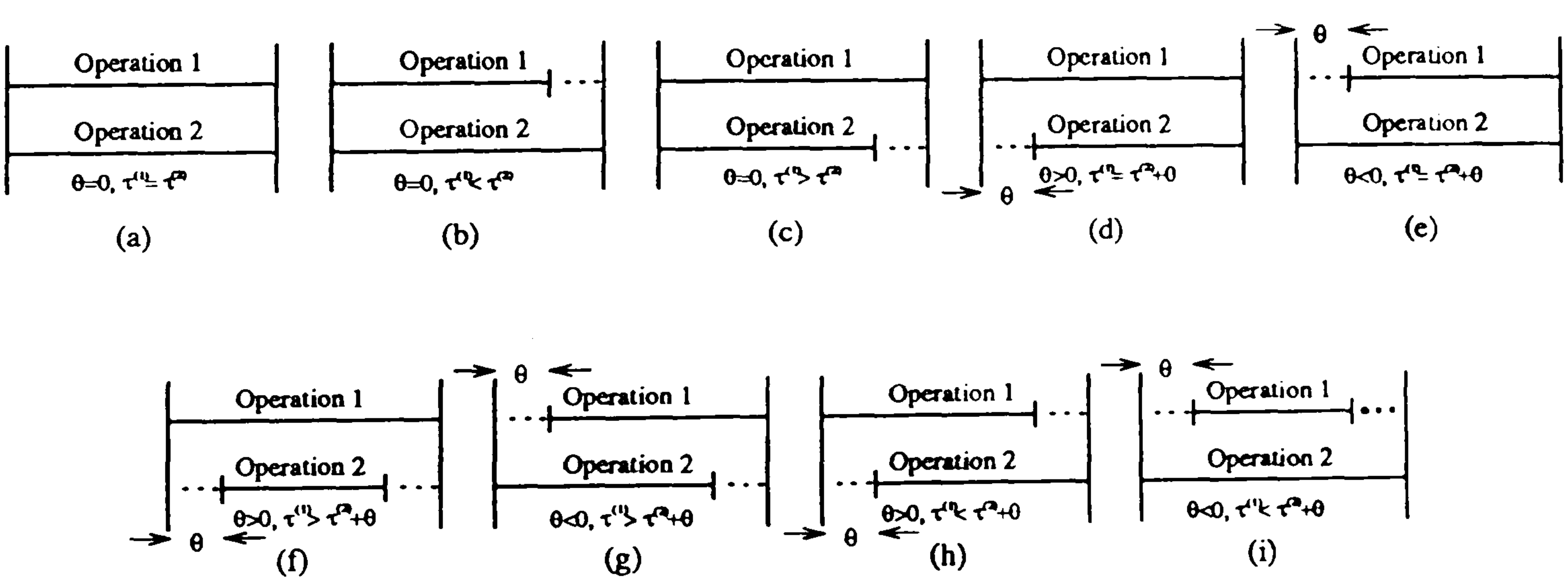


Figure 7.1: Relative Time Arrangements of Energy-integrated Operations

Furthermore, the frequency with which batches can be produced is the same, with the corresponding common cycle time τ_C being bounded from below by the duration of the longer of the two tasks (including any set-up time):

$$\tau_C \geq \tau^{(k)} + \bar{\tau}^{(k)}; \quad k = 1, 2 \tag{7.9}$$

Overall, the optimisation problem to be solved is defined as:

$$\max_{\theta,\, u^{(k)}(t),\, \tilde{u}^{(k)}(t),\, v^{(k)},\, \tau^{(k)};\; k=1,2} \frac{\sum_{k=1}^{2} \phi^{(k)}\left(x^{(k)}(\tau^{(k)}),\, \dot{x}^{(k)}(\tau^{(k)}),\, y^{(k)}(\tau^{(k)}),\, u^{(k)}(\tau^{(k)}),\, \tilde{u}^{(k)}(\tau^{(k)}),\, v^{(k)}\right)}{\tau_C} \tag{7.10}$$

The existence of material coupling between the two operations under consideration introduces additional constraints in the optimisation. For instance, if the output of the first stage forms the feed to the second one, then constraints relating the *initial* values of the variables in the second stage (*i.e.* $x^{(2)}(0)$ and $y^{(2)}(0)$) to the *final* values of the variables in the first stage (*i.e.* $x^{(1)}(\tau^{(1)})$ and $y^{(1)}(\tau^{(1)})$) must be formulated. These can readily be decoupled into initial and end-point equality constraints (of the form (7.2) and (7.3a) respectively) through the introduction of additional time invariant parameters $v^{(k)}$.

Further restrictions, such as zero-wait (ZW) operation between the two processing steps, can be accommodated by the introduction of timing constraints, the form of which depends on which of the two tasks is longer. Thus, if the first task is longer than the second one, it can be shown (see figure 7.2) that the ZW constraint is equivalent to forcing their starting times to coincide (*i.e.* setting $\theta = 0$) in the above optimisation problem.

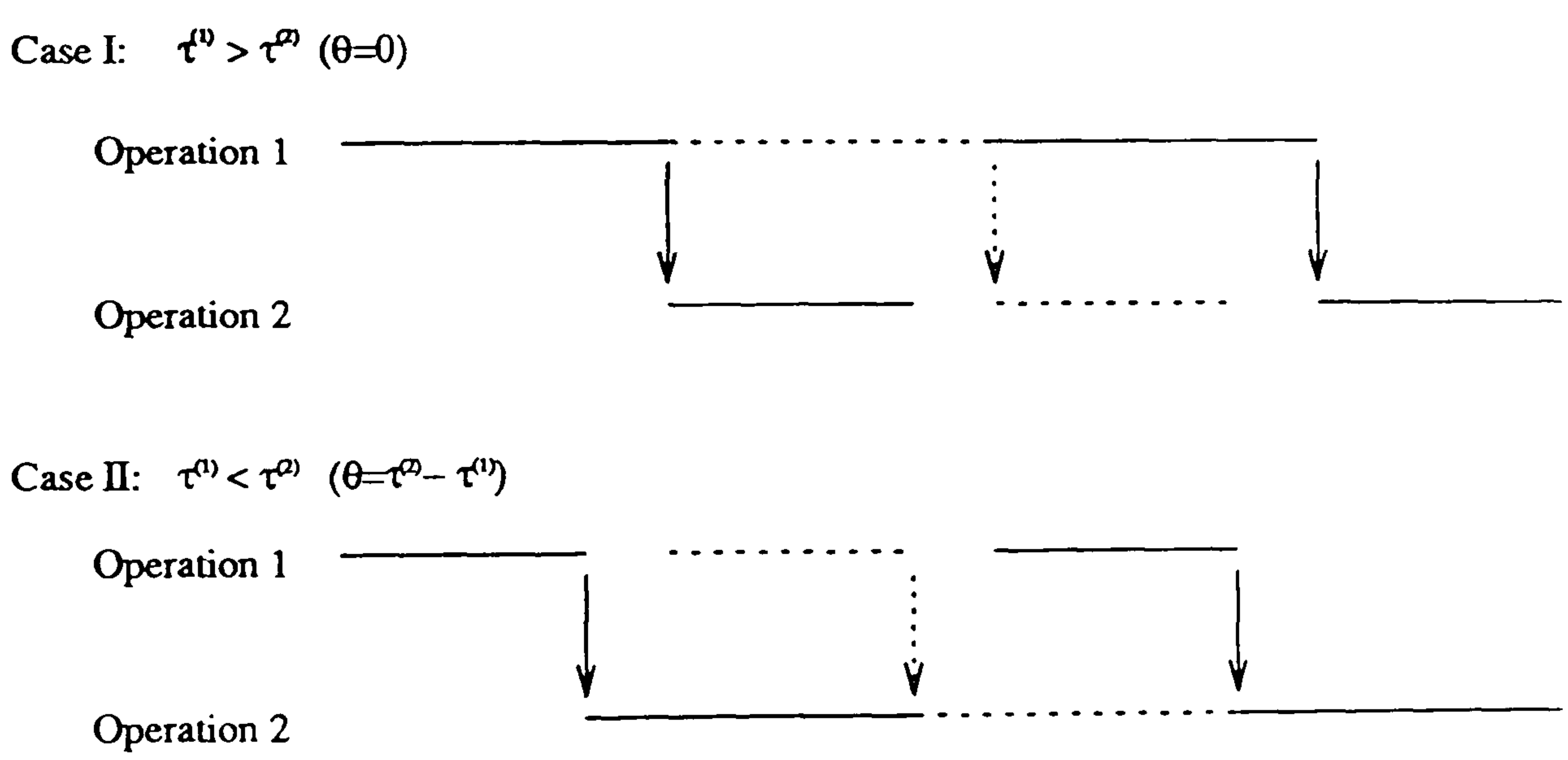


Figure 7.2: Schedules for Zero-wait Operation

On the other hand, if the second task is longer, then the task finishing times must coincide, *i.e.* $\tau^{(2)} = \tau^{(1)} + \theta$. Of course, if the relative magnitude of the two durations is not known *a priori*, it may be necessary to solve *two* optimisation problems, one for each case, and then select the solution with the higher objective function value. The additional necessary timing constraints are

$$\tau^{(1)} \geq \tau^{(2)}\,; \quad \theta = 0 \tag{7.11a}$$

for the first case, and

$$\tau^{(2)} = \tau^{(1)} + \theta\,; \quad \theta \geq 0 \tag{7.11b}$$

for the second, corresponding to arrangements (c) and (e) of figure7.1 respectively.

7.3 Solution Approach

The mathematical optimisation problem defined in the previous section is a multistage optimal control problem. Its complexity primarily arises from the fact that the two operations do not, in general, coincide in time, and therefore the sets of variables and equations describing the combined system vary over the time horizon of interest. Furthermore, it is not known *a priori* which sets of equations and variables are actually active at any given time. This latter complication poses severe problems for currently available optimal control codes.

As shown in figure 7.1, there are nine possible relative arrangements of the two operations in time. However, the first five can be viewed as special cases of the last four by allowing the optimisation procedure to set the idle times in the latter to zero, if this is judged to be advantageous. In principle, then, one could obtain the optimal solution by solving four separate optimisation problems, and selecting the one that gives the highest value of the objective function.

Nevertheless, in practice, it is often possible to exclude some of the possible time arrangements from further consideration on physical grounds. For instance, in the case of the thermally coupled exothermic reaction/distillation process, arrangements (g) and (i)[1] of figure 7.1 would be unlikely to be optimal if it is known *a priori* that the reactor contents require some time before they reach a temperature at which reaction takes place at a non-negligible rate.

The solution of the multistage optimal control problem using standard packages is further facilitated if it is possible to place each of the two operations involved in an "idle" mode by a suitable control action. For instance, such an idle state for batch distillation operations would correspond to total reflux conditions, while for many reactions it can be attained by reducing the reactor temperature to a sufficiently low level. It is then possible to formulate the multistage optimal control problem as a *single-stage* one in which the same sets of equations and variables are active throughout the time horizon, with parts of the latter involving enforced idling in one of the two operations.

The technique outlined above will be clarified by the examples presented in the next section. The problems were modelled within a prototype software package for process

[1] Here we identify operation 1 with the reaction step, and 2 with the distillation.

optimisation (Smith, 1992) acting as a high-level front end to the DAEOPT optimal control code developed by Vassiliadis (1993).

7.4 Examples

7.4.1 Example 1 — Heat Integration without Material Coupling

We consider the thermally coupled reaction-distillation system shown in figure 7.3. The second-order exothermic reaction $A + B \rightarrow C + D$ takes place in a well stirred, liquid phase reactor. The distillation column comprises 5 trays, a reboiler and a total condenser, and is used to carry out a binary separation. It is modelled assuming constant liquid hold-ups in the trays and the condenser, negligible vapour hold-up, and fast energy dynamics. Set-up times of 300 s are assumed for both processing steps.

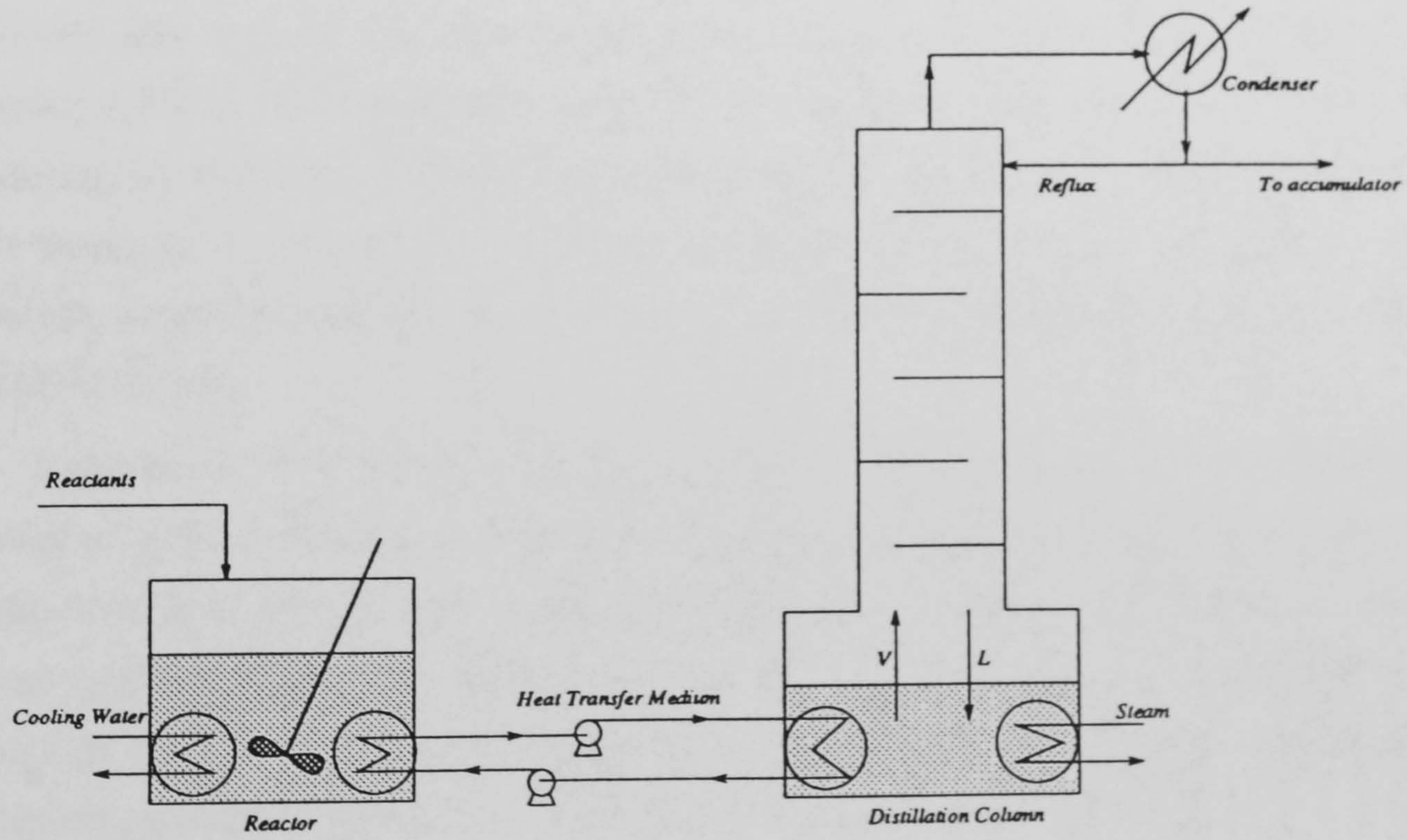


Figure 7.3: Reaction-Distillation Configuration without Material Coupling

The reactor and the column reboiler are coupled through a heat-transfer medium, as well as being able to use supplementary external cooling and heating utilities respectively. The heat-transfer jackets are assumed to be well-stirred vessels with negligible dynamics.

The dynamic models of the two batch operations are presented in detail in Appendix D.

A minimum purity requirement of 97.5% and a minimum recovery of 50% of the more volatile component is imposed on the distillate fraction collected in the accumulator. In order to avoid undesirable side-reactions, the reactor temperature should never exceed 460 K. Furthermore, the reaction is to be quenched by reducing the final reactor temperature to 298 K.

The objective of the optimisation is the maximisation of the rate of value-added of the process taking account of the value of the products, raw materials and external utilities. The control variables that can be manipulated are the flowrate of heat transfer medium, the reflux ratio in the column, and the external cooling and heating loads to the reactor and the column reboiler respectively. Upper bounds are also imposed on some of the control variables; these are 25 kg/s for the heat transfer medium flowrate, 350 kW for the cooling load, and 300 kW for the heating load.

The time horizon of interest is divided into 6 intervals, with piecewise constant control profiles being considered for all control variables except the heat transfer medium flowrate which is assumed to be piecewise linear. The lengths of the intervals, as well as the length of the entire horizon are variables to be determined by the optimisation. Preliminary dynamic simulations indicated that maximum exploitation of heat integration is likely to be achieved by a relative time arrangement of type (h) in figure 7.1. Therefore, the reactor is forced to be effectively idle over the last interval by disallowing any cooling during this interval while enforcing a 298 K final temperature of the reactor contents. Also the column is forced to be idle during the first interval by placing it in a total reflux mode, whilst ensuring that the cost of the steam used in the reboiler during this period is *not* included in the objective function. Moreover, in our calculations, heat exchange is allowed to take place during the second and third intervals only.

A mixture of 10 *kmol* of A and 10 *kmol* of B is charged into the reactor. An amount of 20 *kmol* of a binary mixture containing 80% of the more volatile component is also loaded into the distillation column reboiler. The price of all raw materials is taken to be 0.1 *rcu/mol*[2]. The value of the products is assumed to be 1 *rcu/mol* of C or D produced in the reactor, and 1 *rcu/mol* of top product collected in the column accumulator. Finally, the cost of external heating and cooling utilities is taken to be 0.01 and 0.001 *rcu/kJ* respectively.

The optimal durations of the reaction and distillation steps are found to be 3125 s and 2190 s respectively, leading to a cycle time of 3425 s. The optimal time offset (*i.e.* the length of the first time interval during which the distillation is placed in idle mode) is found to be essentially zero. Thus, the two operations are started almost simultaneously. The temperature profiles for the two operations are shown in figure 7.4a, while figure 7.4b, c, d show optimal profiles of the control variables[3].

---

[2] $rcu \equiv$ relative cost units.

[3] The quantity plotted in figure 7.4b and 7.5b is actually $\hat{r} \equiv r/(1+r)$ where r is the reflux ratio. Thus $\hat{r} = 1$ corresponds to total reflux conditions (*i.e.* $r = \infty$).

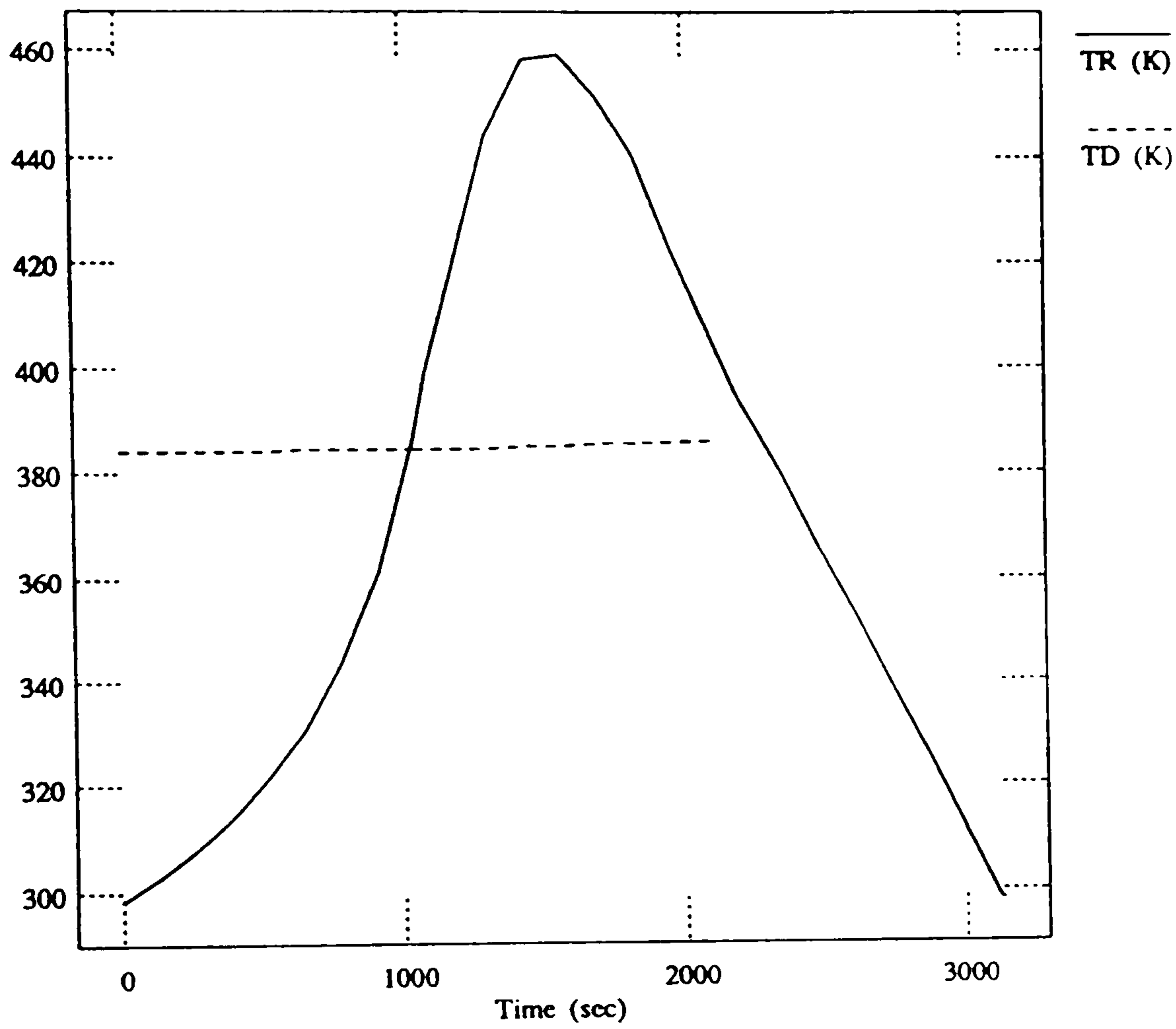


Figure 7.4a: Example 1 — Temperature Profiles (Reaction-TR, Distillation Reboiler-TD)

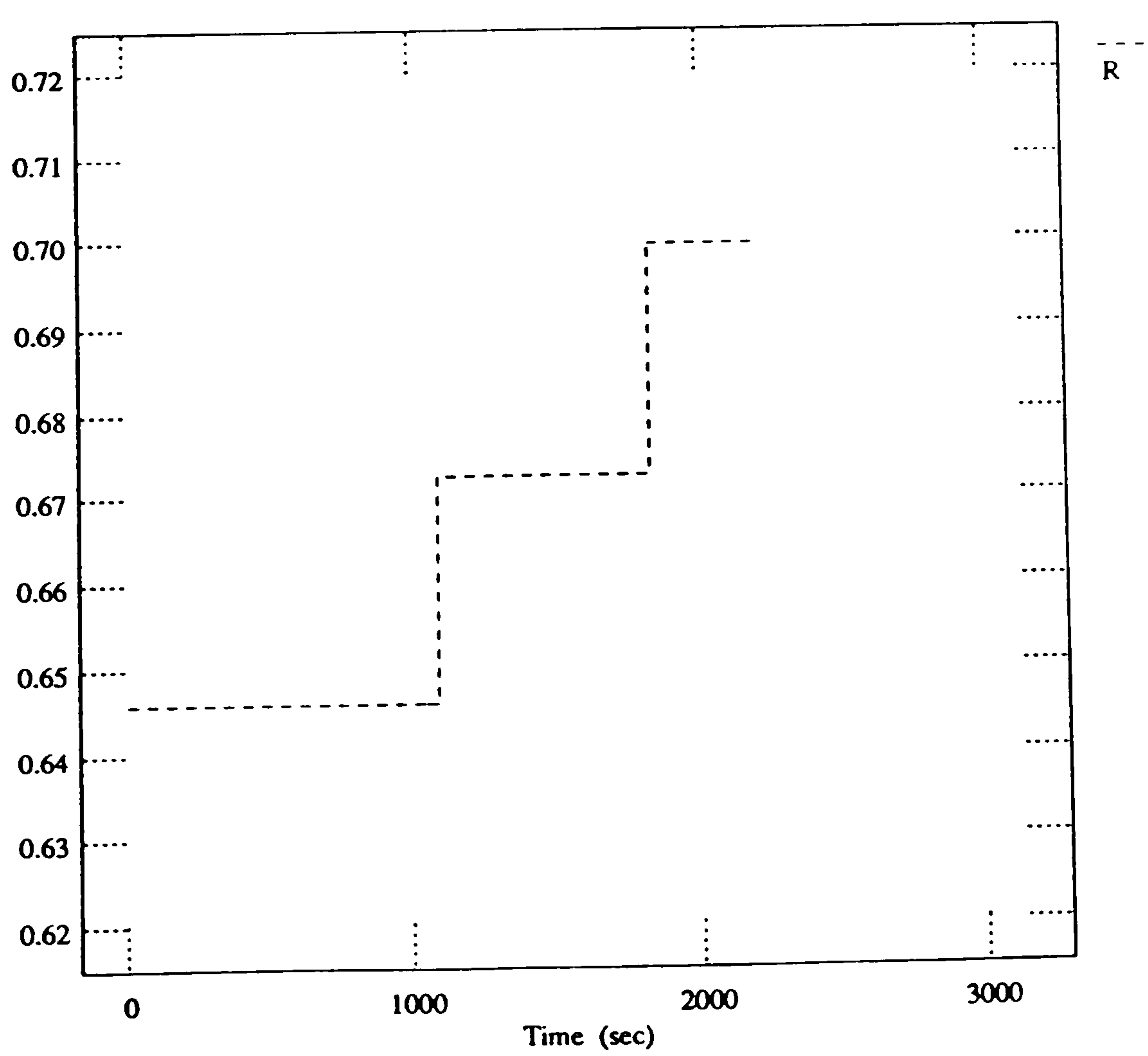


Figure 7.4b: Example 1 — Reflux Ratio

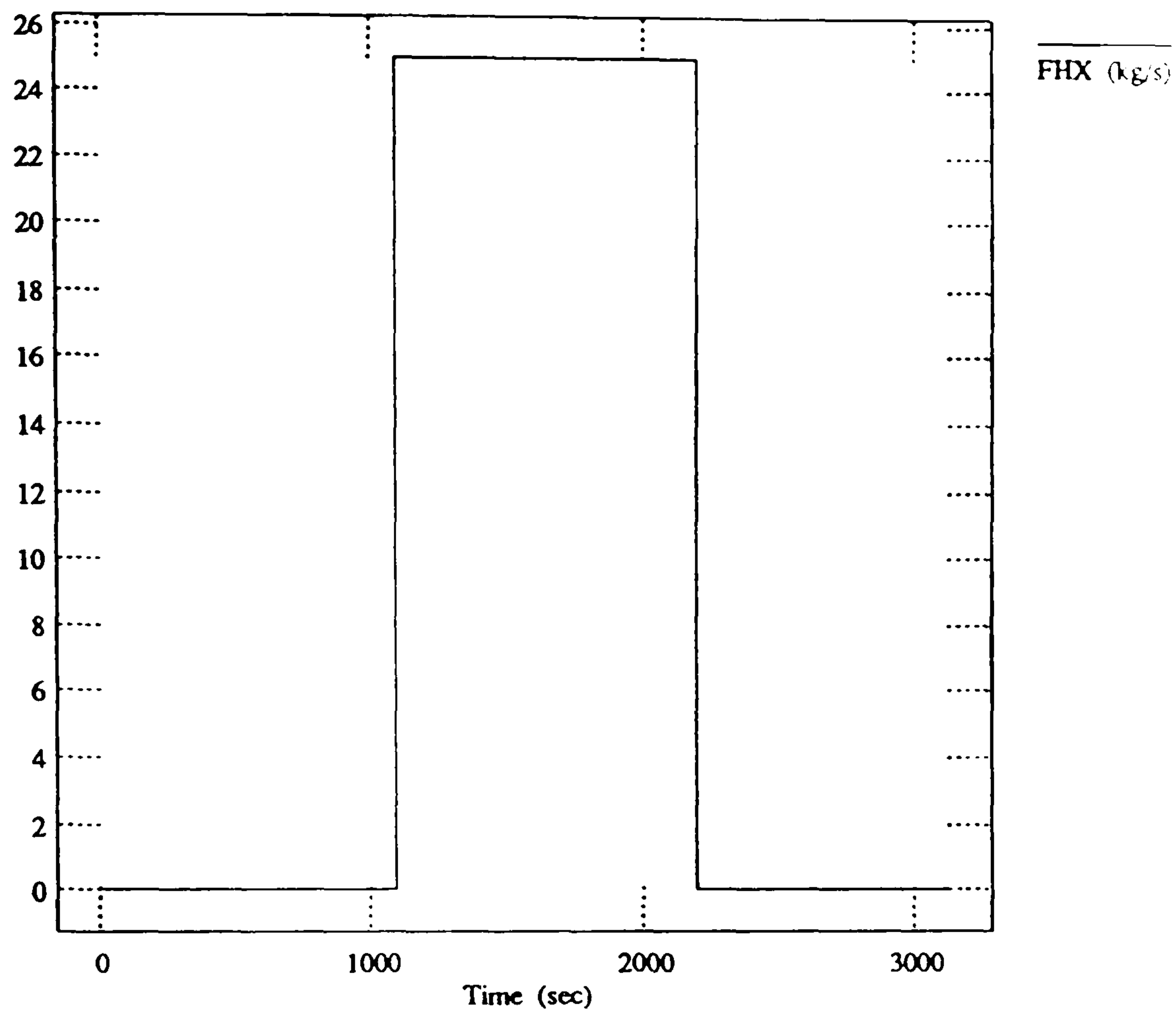


Figure 7.4c: Example 1 — Heat Transfer Medium Flowrate

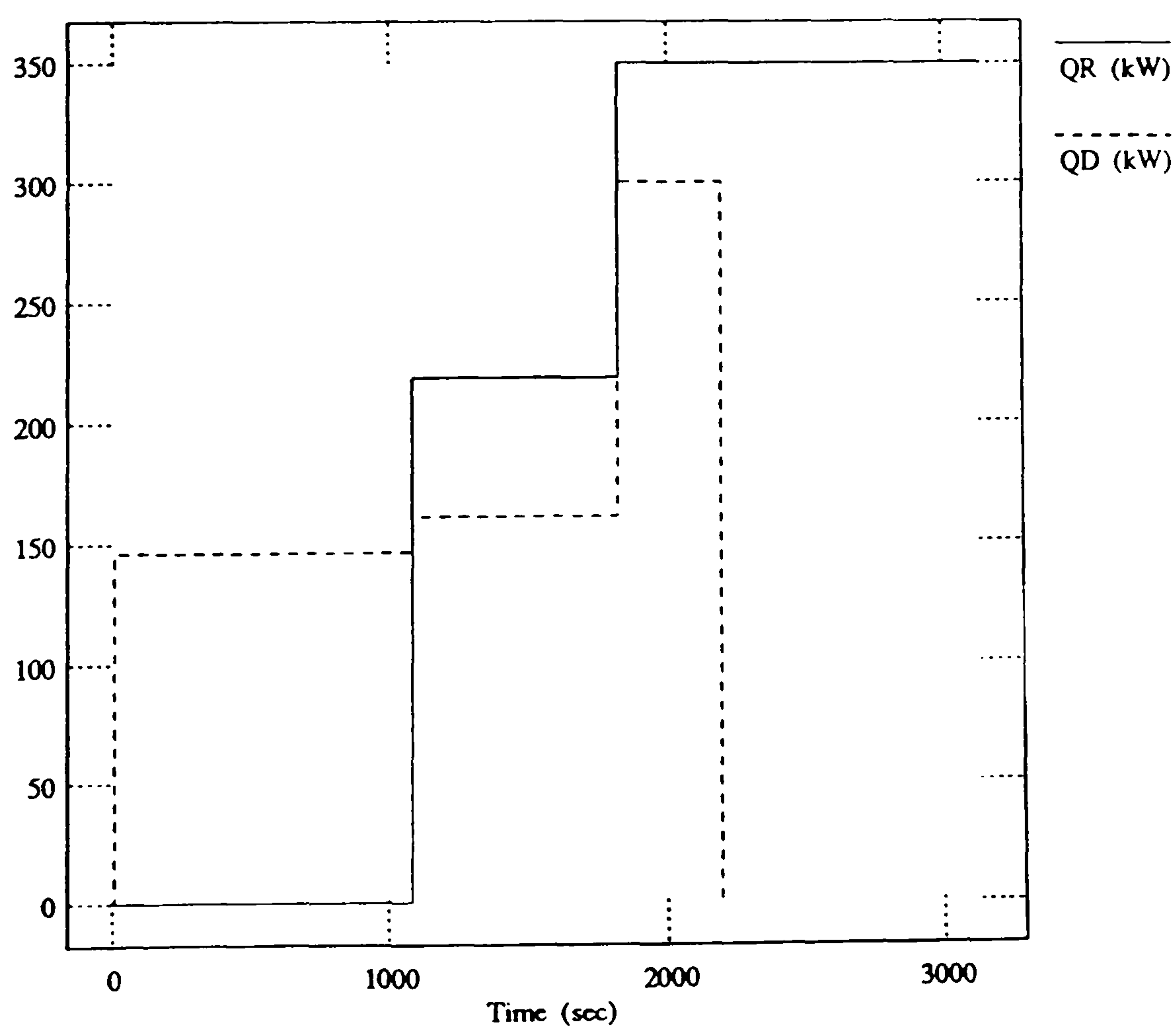


Figure 7.4d: Example 1 — External Utility Provisions (Cooling Water-QR, Steam-QD)

Note that the value of the flowrate of the heat transfer medium (see figure 7.4c) is at its upper limit throughout the period over which heat integration takes place. Supplementary external cooling and heating are also required during this period.

Each batch undergoing reaction consumes 615 MJ of external cooling, while each distillation batch requires 388 MJ of external heating. The corresponding figures for the *unintegrated* operations are 895 and 638 MJ. Thus significant economies in the utilisation of utilities are achieved. Note that the discrepancy in the savings in cooling and heating utilities (280 and 250 MJ per batch respectively) is primarily due to differences in the extent of the reaction taking place, and the amount of distillate collected. The overall objective function values are 4.91 rcu/s for the heat-integrated case, and 3.71 rcu/s for the unintegrated case, *i.e.*, a 32% improvement in economic performance is achieved through energy integration.

7.4.2 Example 2 — Heat Integration with Material Coupling

We now turn our attention to the case in which the product of the reactor is fed to the distillation column for separation. We consider an exothermic second-order reaction of the form $2A \rightarrow B$, with equipment characteristics and modelling assumptions similar to those of the first example. Furthermore, we assume that sufficient storage capacity is available for holding the reaction product waiting to be separated. A maximum temperature of 440 K is imposed on the reaction. Figure 7.5 shows the modified reaction-distillation configuration involving material transfer from the first processing step to the second one.

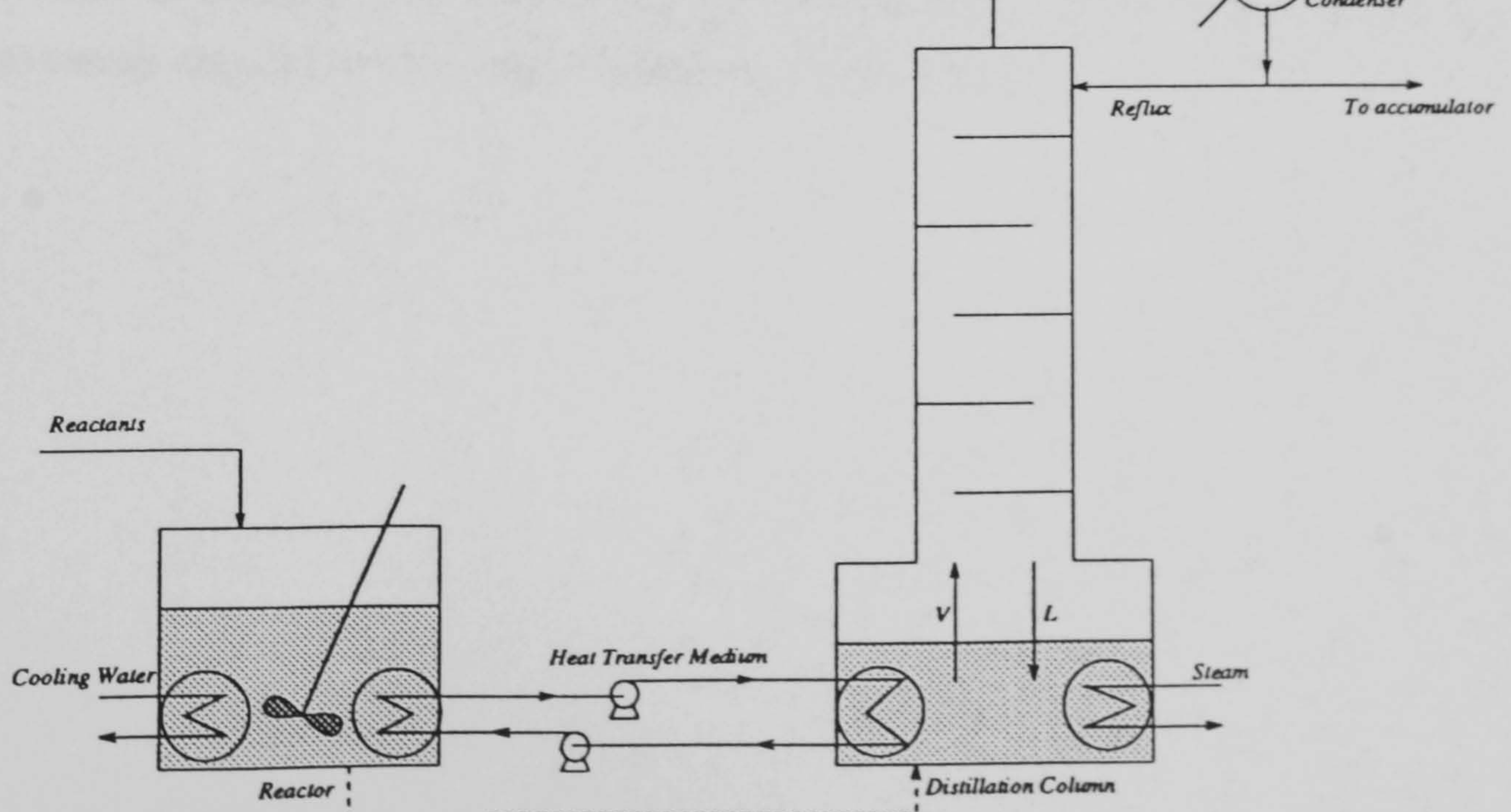


Figure 7.5: Reaction-Distillation Configuration with Material Coupling

In this case, we divide the time horizon into 5 intervals of variable duration, with the reaction and distillation operations being forced to be idle during the last and first intervals respectively. Furthermore, heat exchange is allowed in the third interval only. In addition to the control variables considered in the previous example, the optimisation has to determine the final conversion of component A in the reactor. Mathematically, this is treated as a time-invariant parameter related to the final values of the variables in the reaction operation through an end-point constraint. The initial conditions (*i.e.*, composition) in the column are also expressed in terms of this parameter. Clearly, higher conversions would tend to facilitate the separation task at the expense of longer reaction times, and this decision also interacts strongly with the external utility requirements.

The initial batch fed to the reactor consists of 20 *kmol* of pure A at a cost of 0.1 *rcu/mol*. The value of the final product collected in the column accumulator is assumed to be 2 *rcu/mol*.

The optimal duration of the reaction is found to be 4360 s, corresponding to a 95.5% conversion of A. The distillation operation starts with a time offset of 1150 s and has a duration of 3770 s. Thus, the overall cycle time is 4660 s while the non-zero offset implies that the intermediate storage is indeed necessary for holding the reaction product temporarily before it can be charged into the column.

Figures 7.6a-7.6d shows the temperature and control variable profiles for the two operations. It can be seen from figure 7.6a and 7.6c that heat exchange takes place for a period of 1580 s during which the temperature of the reactor exceeds that of the reboiler. Figure 7.6d shows that the cooling load is set to its maximum value for two periods during the horizon: in the first, this aims primarily at preventing the reactor temperature from exceeding the limit of 440 K while in the second the main objective is to quench the reaction as quickly as possible so as to ensure shorter reaction, and consequently cycle times.

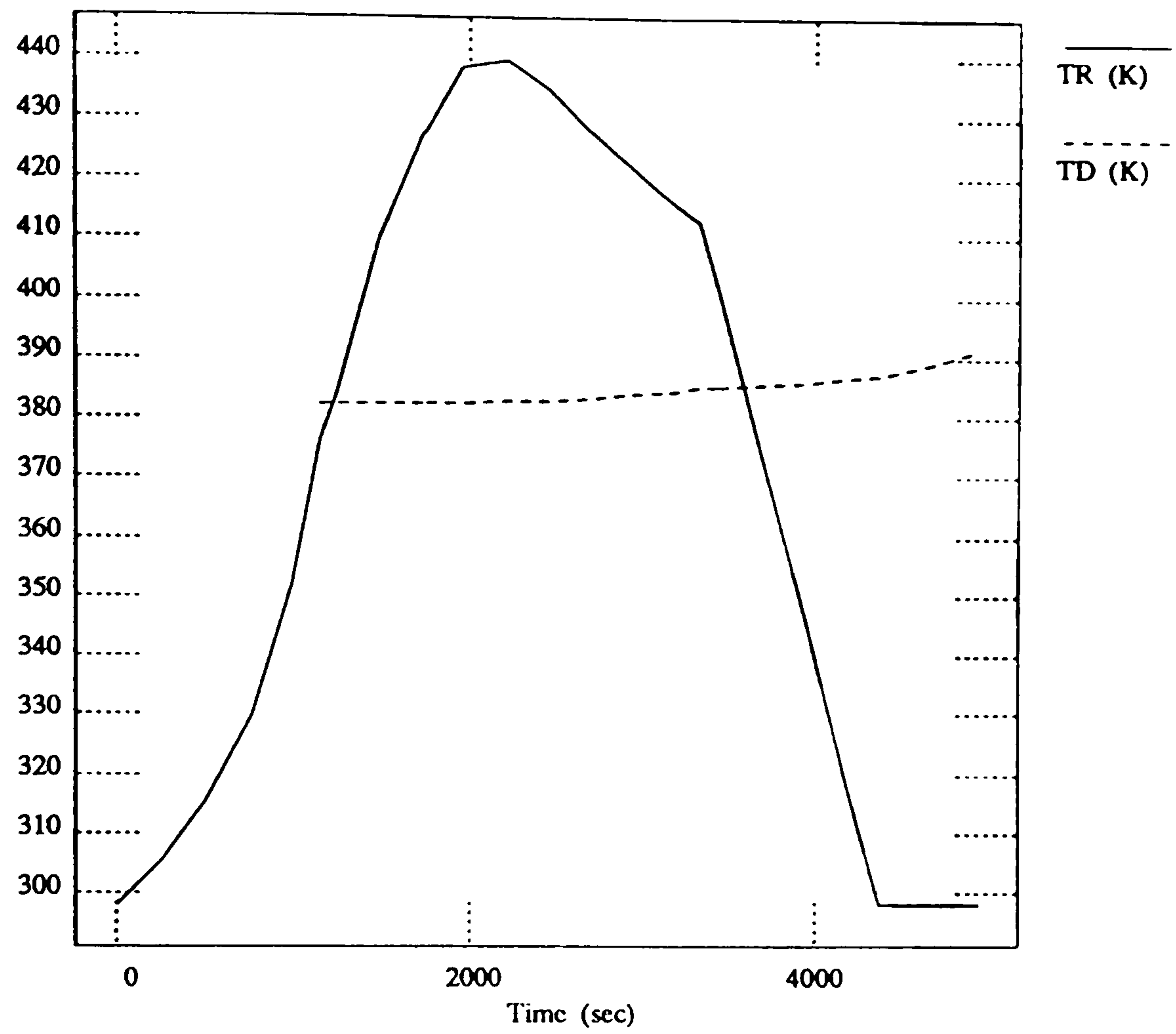


Figure 7.6a: Example 2 — Temperature Profiles (Reaction-TR, Distillation Reboiler-TD)

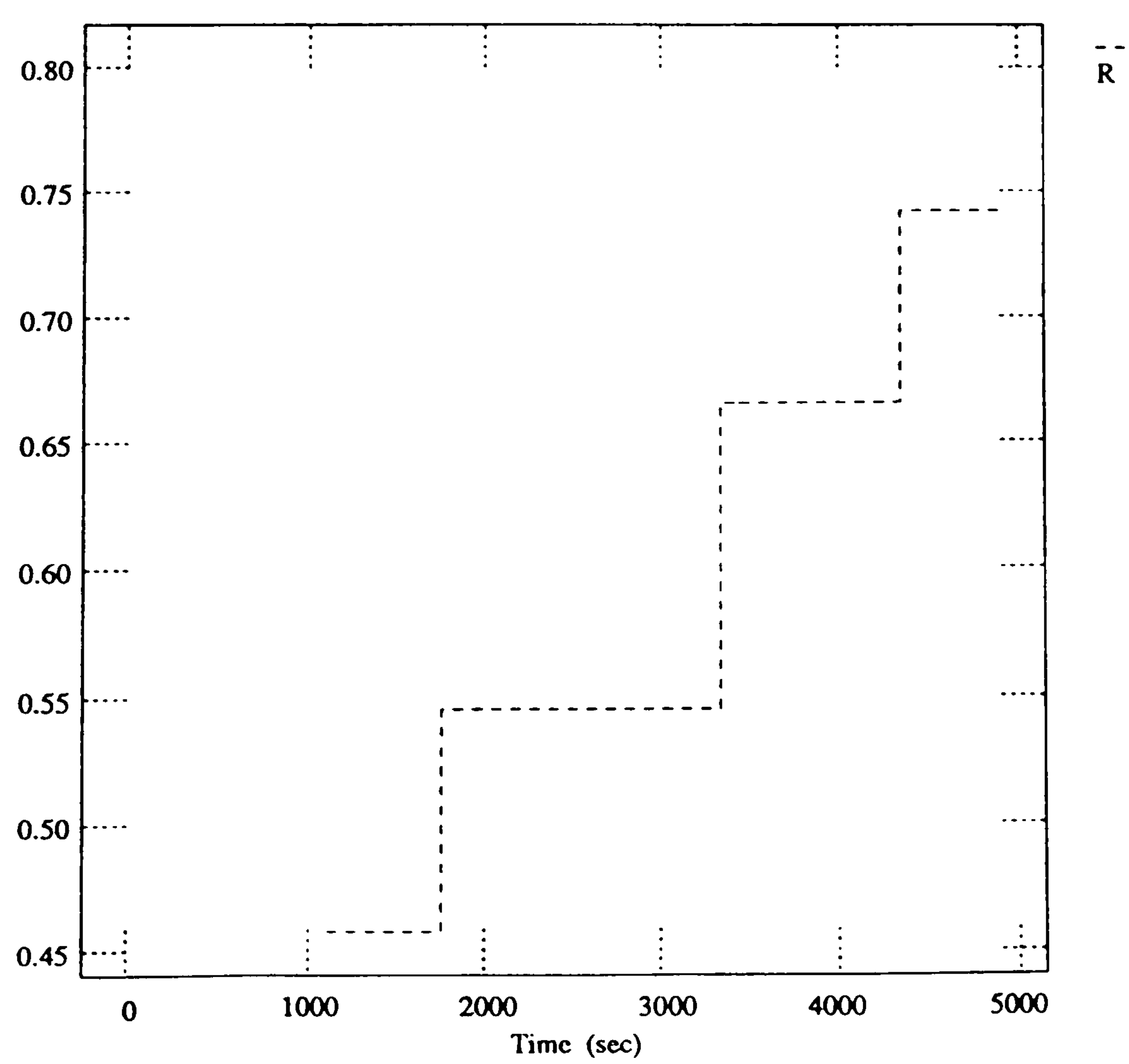


Figure 7.6b: Example 2 — Reflux Ratio

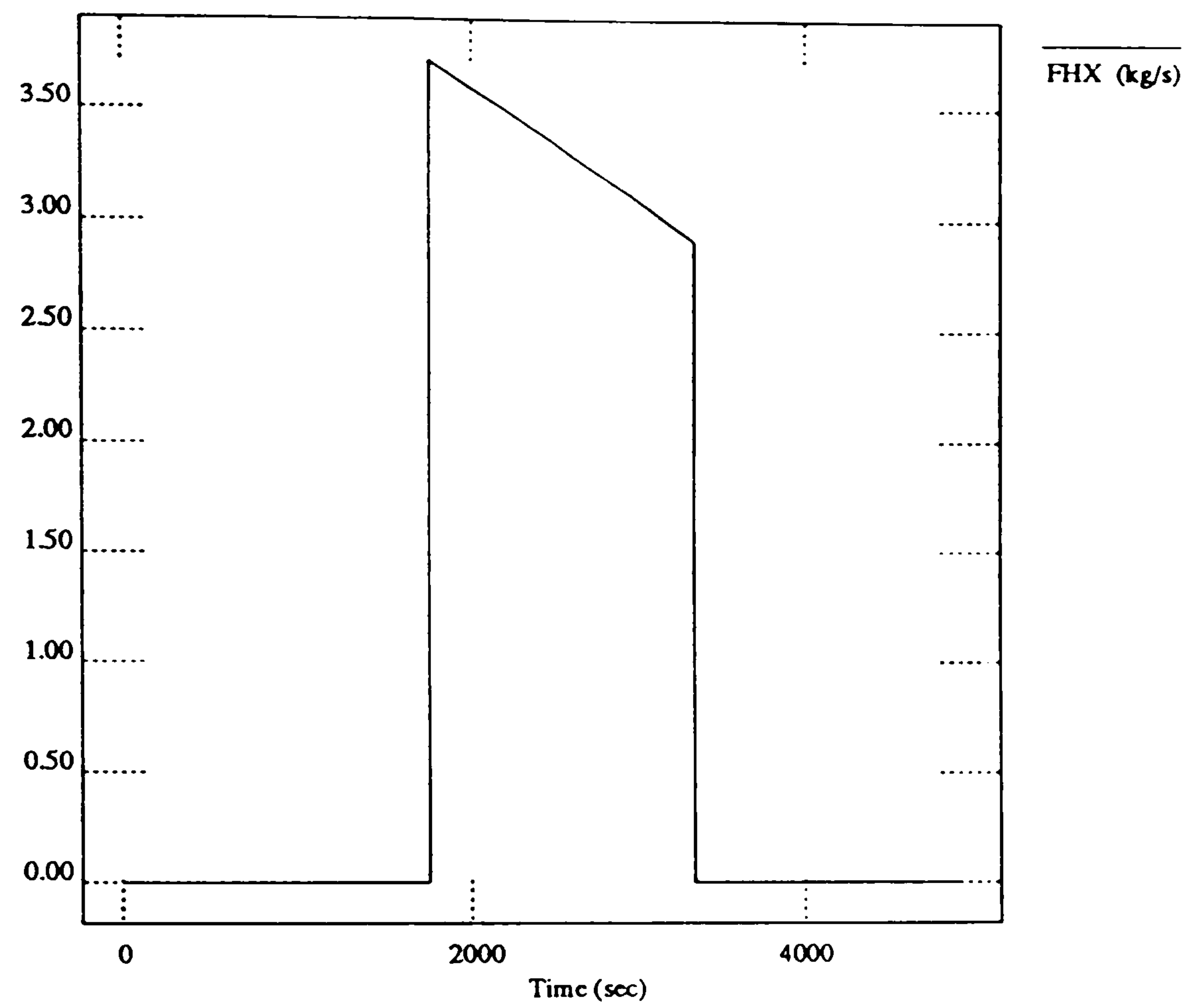


Figure 7.6c: Example 2 — Heat Transfer Medium Flowrate

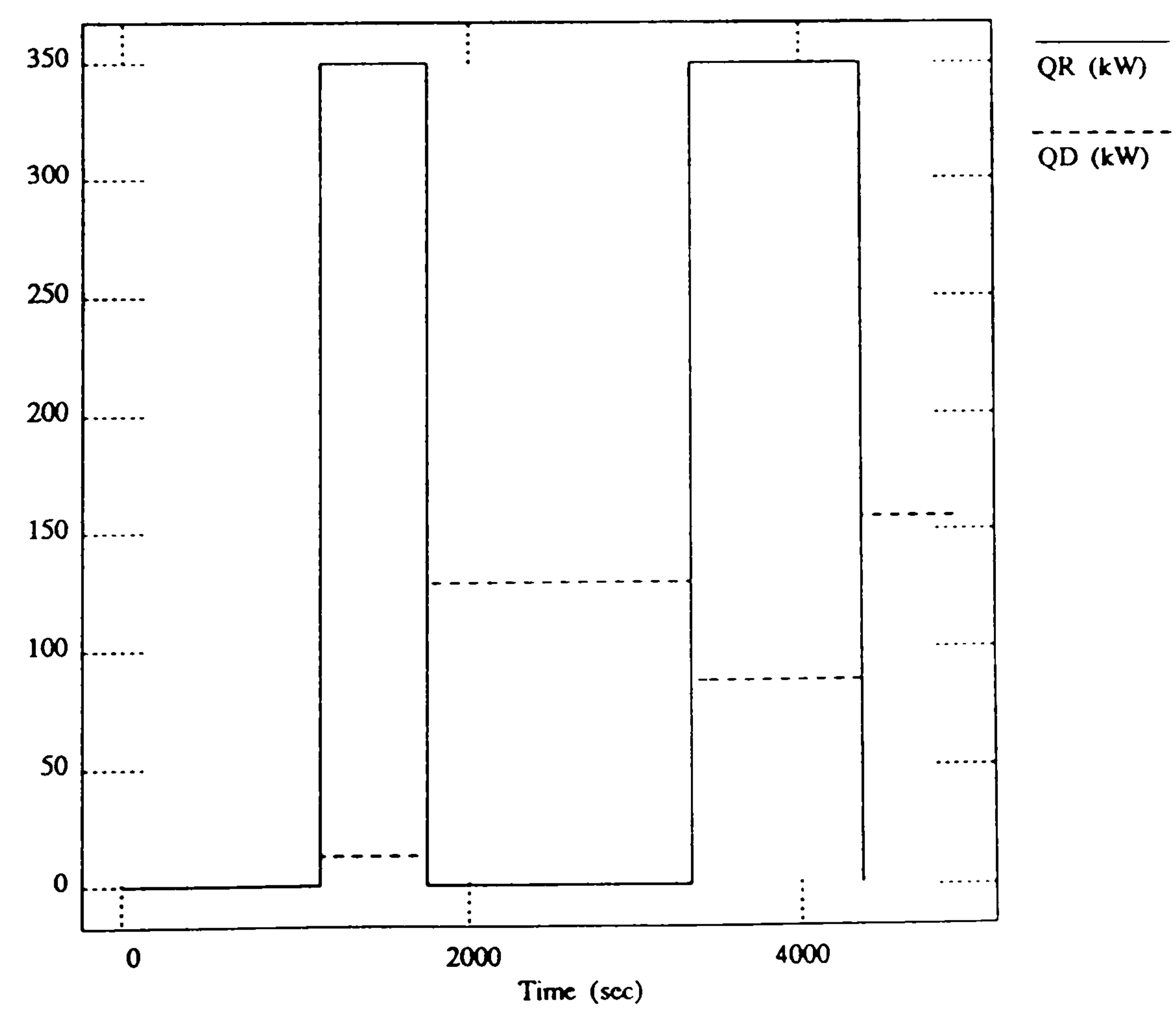


Figure 7.6d: Example 2 — External Utility Provisions (Cooling Water-QR, Steam-QD)

The cooling and heating requirements per batch are found to be 570 *MJ* and 385 *MJ* respectively. The corresponding figures for the same system with material coupling but without heat integration are 778 *MJ* and 566 *MJ* respectively, corresponding to a 95.7% conversion of *A*. The values of the overall objective function with and without energy integration are respectively 2.60 *rcu/s* and 2.22 *rcu/s*, *i.e.* energy integration results in a 17% improvement in economic performance.

7.5 Conclusions

This chapter has studied the rigorous optimisation of pairs of energy integrated batch operations with or without direct material coupling. The approach advocated recognises and takes account of the interactions between energy integration and other available control manipulations, as well as the effects that these may have on product quality and quantity.

The solution technique presented converts the multistage optimal control problem to a single stage one through the introduction of idle periods of operation of the individual units. Physical reasoning may be used to decide where these idle periods should be placed, but their precise duration is determined automatically. In any case, a more direct approach to the solution of the multistage problem itself could be both more efficient and easier to apply.

It should be noted that the attention in this chapter was focussed entirely on the optimisation of closely coupled and (at least partially) overalapping batch operations, with all heat exchange taking place during the period of overlap. This, in a sense, is the more interesting case as it involves the strongest interactions between the two batch operations. In principle, optimal control techniques similar to those presented here could also be applied to indirectly coupled batch operations where a storable heat transfer medium serves to provide a degree of temporal decoupling.

Chapter 8

Conclusions and Future Directions

This thesis has been concerned with two distinct areas of optimal operation of multipurpose batch/semi-continuous plants, namely campaign planning and scheduling and energy integration. The key contributions in each of them are summarised in the next two sections, while section 8.3 considers possible directions for future work.

8.1 Campaign Planning/Scheduling

Earlier mathematical programming-based formulations presented in the literature dealt with the campaign planning problems of simplified processing recipes (*e.g.* serial networks), with various restrictions on resource utilisation (*e.g.* equipment pre-assignment of processing tasks, no equipment re-use during each campaign, ZW mode of operation, in-phase and out-of-phase operation of parallel equipment *etc*.).

In chapter 3, we presented a mathematical formulation of the combined campaign planning/scheduling problem. The key features distinguishing this work are:

- generality with respect to process structure;
- flexibility with respect to the utilisation of processing and storage equipment;
- consideration of limited availability and cost of utilities;
- generality in the supply of raw materials and demand for each product; and
- seasonal variations in the problem data.

The problem is formulated as a large mixed integer linear programme. Techniques for reducing the search space associated with the solution of this MILP were considered in chapter 4. Finally, chapter 5 presented a mathematical decomposition approach for the solution of the planning/scheduling problem. It should be emphasised that the techniques of chapters 4 and 5 are entirely complementary.

The campaign planning/scheduling formulation presented here is entirely consistent with the STN framework of multipurpose plant optimisation (Pantelides, 1994). It has already been implemented within the *gBSS* package (Papageorgiou *et al.*, 1992) complementing earlier facilities for short-term and cyclic scheduling, and plant design.

8.2 Energy Integration

The exploitation of energy integration opportunities in the operation of multipurpose batch/semi-continuous plants has been examined in chapters 6 and 7.

The overall problem has been decomposed into two subproblems. At the higher level, the optimal scheduling of these plants, considered in chapter 6, takes into account simultaneously both stand-alone and heat-integrated operations so as to optimise an economic performance criterion (*e.g.* minimisation of external utilities cost).

Two modes of energy integration were considered; direct and indirect ones. In direct mode, the coupled processes exchange energy simultaneously while in indirect mode, the energy is transferred via a heat transfer medium among various processes allowing the possibility of storing energy over certain periods of time, thus increasing the operational flexibility of the coupled tasks. This categorisation does not preclude the co-existence of both modes within the same plant.

An important consideration in the indirect energy integration case is the variation over time of the temperature of the heat transfer medium kept in storage. This work has extended the concept of the STN to allow states to be characterised by intensive variables (temperatures) as well as extensive ones (inventories). The mathematical formulation presented in chapter 6 is based on a detailed characterization of the variation of the mass and energy holdups of HTM over time. In particular, it takes account of the limitations on the duration of energy storage due to heat losses to the environment. A modified branch-and-bound procedure is proposed for the solution of the resulting nonconvex mixed integer nonlinear programming problem (MINLP).

One fundamental assumption in the above work is that the operating characteristics of the heat-integrated tasks are assumed to be known. These include their duration, the external utilities provision, the heat load to be received from heat integration, and the time offset with respect to the starting times of the coupled tasks (direct mode). This information is generated separately at the lower level of the proposed approach where the optimal operation of energy coupled tasks is examined in detail, as described in chapter 7. Detailed dynamic models of the transient behaviour of the energy coupled tasks are employed, and the problem is formulated as a multistage optimal control problem.

8.3 Perspectives for Future Work

This thesis has been concerned with aspects of the *operation* of multipurpose plants. It is clear, however, that there are corresponding *design* problems concerned, for instance, with:

- the design of plants operating in campaign mode;
- the design of energy integrated plants, considering the selection of the main equipment and the configuration of the utility and heat transfer medium networks; and
- the detailed design and sizing of energy-integrated equipment units.

In the first two cases, there are obvious parallels with the earlier work of Shah (1992) and Barbosa-Póvoa (1994). All these contributions are based on the discrete time STN framework, and this consistency should facilitate progress in this area.

The detailed design of energy-integrated process equipment is another dynamic optimisation problem with the design characteristics treated as additional time-invariant parameters to be determined by the optimisation.

The decomposition approach used for the solution of the combined planning/scheduling problem has been found to be very effective requiring a relatively small number of iterations between master problem and subproblem. The use of better aggregate models (taking, for instance, better account of equipment cleaning requirements) would lead to further improvements in this area.

The extension of the concept of "state" to involve intensive as well as extensive properties opens some interesting possibilities. For instance, the introduction of composition variables would allow more general task models to be used. Some practical processes (*e.g.* blending/splitting) could be described in terms of purely *algebraic* models, and in these cases the techniques developed in chapter 6 for dealing with non-convex MINLPs may be useful. However, for more complex operations, the use of *differential/algebraic* models may be inevitable (Charalambides *et al.*, 1993).

References

Barbosa-Póvoa, A.P.F.D., "*Detailed Design and Retrofit of Multipurpose Batch Plants*", PhD Thesis, University of London (1994).

Birewar, D.B., and I.E. Grossmann, "Incorporating Scheduling in the Optimal Design of Batch Processes", *Comput. chem. Engng.*, **13**, 141-161 (1989a).

Birewar, D.B., and I.E. Grossmann, "Efficient Optimization Algorithms for Zero-Wait Scheduling of Multiproduct Batch Plants", *Ind. Eng. Chem. Process Des. Dev.*, **28**, 1333-1345 (1989b).

Birewar, D.B., and I.E. Grossmann, "Simultaneous Production Planning and Scheduling in Multiproduct Batch Plants", *Ind. Eng. Chem. Res.*, **29**, 570-580 (1990).

Charalambides, M.S., N. Shah, and C.C. Pantelides, "Optimal Batch Process Synthesis", Paper presented at AIChE Annual Meeting, St Louis, USA (Nov, 1993).

Corominas, J., A. Espuña, and L. Puigjaner, "A New Look at Energy Integration in Multiproduct Batch Processes", *Comput. chem. Engng.*, **17S**, S15-S20 (1993).

Crooks, C.A., "*Synthesis of Operating Procedures for Chemical Plants*", PhD Thesis, University of London (1992).

Drud, A.S., "*CONOPT — A GRG code for Large-Scale Nonlinear Optimization — Reference Manual*", ARKI Consulting and Development A/S, Bagsvaerd, Denmark (1992).

Glover, F., "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems", *Man. Sci.*, **22**, 455 (1975).

Hasebe, S., and I. Hashimoto, "Present Status of Batch Process Systems Engineering in Japan", NATO Advanced Study Institute on Batch Processing Systems Engineering, Antalya, Turkey (1992).

Ivanov, B., K. Peneva, and N. Bancheva, "Heat Integration of Batch Vessels at Fixed Time Interval. I. Schemes with Recycling Main Fluids", *Hungarian Journal of Industrial Chemistry*, **20**, 225-231 (1992).

Kemp, I.C., and E.K. MacDonald, "Energy and Process Integration in Continuous and Batch Processes", *IChemE Symposium Series*, **105**, 185-200 (1988a).

Kemp, I.C., and E.K. MacDonald, "Application of Pinch Technology to Separation, Reaction and Batch Processes", *IChemE Symposium Series*, **109**, 239-257 (1988b).

Kemp, I.C., and A.W. Deakin, "The Cascade Analysis for Energy and Process Integration of Batch Processes. Part 1: Calculation of Energy Targets", *Chem. Eng. Res. Des.*, **67**, 495-509 (1989).

Kondili, E., "*Optimal Scheduling of Batch Chemical Processes*", PhD Thesis, University of London (1988).

Kondili, E., C.C. Pantelides, and R.W.H. Sargent, "A General Algorithm for Scheduling Batch Operations", *Proc. Third Int. Symp. on Process Systems Eng.*, 62-75, Sydney, Australia (1988).

Kondili, E., C.C. Pantelides, and R.W.H. Sargent, "A General Algorithm for Short-Term Scheduling of Batch Operations. Part I - Mathematical Formulation", *Comput. chem. Engng.*, **17**, 211-227 (1993).

Linnhoff, B., G.J. Ashton, and E.D.A. Obeng, "Process Integration of Batch Processes", *IChemE Symposium Series*, **109**, 221-237 (1988).

Linnhoff, B., D.W. Townsend, D. Boland, G.F. Hewitt, B.E.A. Thomas, A.R. Guy, and R.H. Marsland, "*User Guide on Process Integration for the Efficient Use of Energy*", IChemE, Rugby (1982).

Lovell-Smith, J.E.R., and A.J. Baldwin, "Energy Saving Opportunities in Dairy Manufacturing Plant", *New Zealand Journal of Dairy Science and Technology*, **23**, 61-74 (1988).

Mauderli, A., and D.W.T. Rippin, "Production Planning and Scheduling for Multi-purpose Batch Chemical Plants", *Comp. chem. Engng.*, **3**, 199-206 (1979).

McCormick, G.P., "Computability of Global Solutions to Factorable Nonconvex Programs: Part I — Convex Underestimating Problems", *Math. Program.*, **10**, 147-175 (1976).

Mignon, D., and J. Hermia, "Using BATCHES for Modeling and Optimizing the Brewhouses of an Industrial Brewery", *Comput. chem. Engng.*, **17S**, S51-S56 (1993).

MPSX, "*v.2 — User Guide*", IBM (1990).

Murtagh, B.A., and M.A. Saunders, "*MINOS 5.0 User's Guide*", Technical Report SOL 83-20, Stanford University (1983).

Nemhauser, G.L., and L.A. Wolsey, "*Integer and Combinatorial Optimization* ", John Wiley and Sons, New York (1988).

Obeng, E.D.A., and G.J. Ashton, "On Pinch Technology Based Procedures for the Design of Batch Processes", *Chem. Eng. Res. Des.*, **66**, 255-259 (1988).

Pantelides, C.C., "Unified Frameworks for Optimal Process Planning and Scheduling", *Proc. Second Conf. on Foundations of Computer-Aided Operations*, Rippin, D.W.T. and Hale, J. eds., CACHE Publications, 253-274 (1994).

Papageorgiou, L.G., and C.C. Pantelides, "A Hierarchical Approach for Campaign Planning of Multipurpose Batch Plants", *Comp. chem. Engng.*, **17S**, S27-S32 (1993).

Papageorgiou, L.G., N. Shah, and C.C. Pantelides, "A Software System for Optimal Planning and Scheduling of Batch Operations", *Proc. Advances in Process Control III*, 161-170, Rugby (1992).

Parakrama, R., "Improving Batch Processes", *The Chemical Engineer*, September, 24-25 (1985).

Peneva, K., B. Ivanov, and N. Bancheva, "Heat Integration of Batch Vessels at Fixed Time Interval. II. Schemes with Intermediate Heating and Cooling Agents", *Hungarian Journal of Industrial Chemistry*, **20**, 233-239 (1992).

Quesada, I., and I.E. Grossmann, "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems", *Comput. chem. Engng.*, **16**, 937-947 (1992).

Reklaitis, G.V., "Perspectives on Scheduling and Planning of Process Operations", *Proc. Fourth Int. Symp. on Process Systems Engng.*, Montreal, Canada (1991).

Reklaitis, G.V., "Overview on Scheduling and Planning of Process Operations," NATO Advanced Study Institute on Batch Processing Systems Engineering, Antalya, Turkey (1992).

Rippin, D.W.T., "Batch Process Systems Engineering: a Retrospective and Prospective Review", *Comput. chem. Engng.*, **S17**, S1-S13 (1993).

SD-SCICON, "*SCICONIC vs. 2.11 — User Manual*" (1990).

Shah, N., "*Efficient Scheduling, Planning and Design of Multipurpose Batch Plants*", PhD Thesis, University of London (1992).

Shah, N., and C.C. Pantelides, "Optimal Long-term Campaign Planning and Design of Batch Operations", *Ind. Eng. Chem. Res.*, **30**, 2308-2321 (1991).

Shah, N., C.C. Pantelides, and R.W.H. Sargent, "Optimal Periodic Scheduling of Multipurpose Batch Plants", *Annals of Operations Research*, **42**, 193-228 (1993a).

Shah, N., C.C. Pantelides, and R.W.H. Sargent, "A General Algorithm for Short-Term Scheduling of Batch Operations. Part II - Computational Issues", *Comput. chem. Engng.*, **17**, 229-244 (1993b).

Smith, E.M.B., "*A Modelling Environment for Process Optimisation*", Technical Report, Imperial College, London (1992).

Tsirukis, A.G., and G.V. Reklaitis, "Feature Extraction Algorithms for Constrained Global Optimization I. Mathematical Foundation", *Annals of Operations Research*, **42**, 229-273 (1993a).

Tsirukis, A.G., and G.V. Reklaitis, "Feature Extraction Algorithms for Constrained Global Optimization II. Batch Process Scheduling Application", *Annals of Operations Research*, **42**, 275-312 (1993b).

Tsirukis, A.G., S. Papageorgaki, and G.V. Reklaitis, "Scheduling of Multipurpose Batch Chemical Plants with Resource Constraints", *Ind. Eng. Chem. Res.*, **32**, 3037-3050 (1993).

Vaselenak, J.A., I.E. Grossmann, and A.W. Westerberg, "Heat Integration in Batch Processing", *Ind. Eng. Chem. Proc. Des. Dev.*, **25**, 357-366 (1986).

Vassiliadis, V., "*Computational Solution of Dynamic Optimization Problems with General Differential-Algebraic Constraints*", PhD Thesis, University of London (1993).

Voudouris, T.V., and I.E. Grossmann, "Optimal Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations", *Ind. Eng. Chem. Res.*, **32**, 1962-1980 (1993).

Wellons, M.C., and G.V. Reklaitis, "Optimal Schedule Generation for a Single-Product Production Line— I. Problem Formulation", *Comput. chem. Engng.*, **13**, 201-212 (1989a).

Wellons, M.C., and G.V. Reklaitis, "Optimal Schedule Generation for a Single-Product Production Line— II. Identification of Dominant Unique Path Sequences", *Comput. chem. Engng.*, **13**, 213-227 (1989b).

Wellons, M.C., and G.V. Reklaitis, "Scheduling of Multipurpose Batch Chemical Plants. 1. Formation of Single-Product Campaigns", *Ind. Eng. Chem. Res.*, **30**, 671-688 (1991a).

Wellons, M.C., and G.V. Reklaitis, "Scheduling of Multipurpose Batch Chemical Plants. 2. Multiple Product Campaign Formation and Production Planning", *Ind. Eng. Chem. Res.*, **30**, 688-705 (1991b).

Williams, H.P., "*Model Building in Mathematical Programming*", John Wiley and Sons, Chichester (1990).

Appendix A

Derivation of Bounds on Nett Material Accumulation per Cycle

Here we consider the derivation of tight bounds on the quantities ΔS_{sc} representing the nett accumulation of material in state s over a cycle of campaign c (*cf.* section 3.2). These bounds are essential for the effective linearisation of bilinear terms in the material balance constraints (see section 3.2.5).

Two alternative ways of deriving such tight upper bounds are described.

A.1 Bounds Based on Task Processing Capacity

The obvious upper bound on ΔS_{sc} is C_s, the dedicated storage capacity for state s. Unfortunately, for some states, this is likely to be much larger than the nett accumulation over any single cycle. However, a much tighter upper bound may be obtained. Summing the material balance constraints (3.28) over time t, we get:

$$S^c_{s,T+1} = S^c_{s0} + \sum_{t=1}^{T_c+1} \left(\sum_{i \in \overline{T}_s} \overline{\rho}_{is} \sum_{j \in K_i} B^c_{ij,\tau(t-p_{is})} - \sum_{i \in T_s} \rho_{is} \sum_{j \in K_i} B^c_{ijt} \right) \tag{A.1}$$

Since $\Delta S_{sc} \equiv S^c_{s,T_c+1} - S^c_{s0}$, we can obtain, by reversing the order of the summations on the right hand side of (A.1):

$$\Delta S_{sc} = \sum_{i \in \overline{T}_s} \overline{\rho}_{is} \sum_{j \in K_i} B^{c\,*}_{ij} - \sum_{i \in T_s} \rho_{is} \sum_{j \in K_i} B^{c\,*}_{ij} \tag{A.2}$$

where $B^{c\,*}_{ij} \equiv \sum_{t=1}^{T_c} B^c_{ijt}$. Note that we have made use of the fact that, because of the cyclic nature of the schedule,

$$\sum_{t=1}^{T_c} B_{ij,\tau(t-p_{is})} = \sum_{t=1}^{T_c} B^c_{ijt} = B^{c\,*}_{ij}.$$

Because of the allocation constraints (3.25), the maximum number of batches of task i that can be performed over the cycle period T_c for an active campaign (*i.e.* $X_c = 1$) is:

$$\left\lfloor \frac{T_c}{p_i} \right\rfloor \geq \sum_{t=1}^{T_c} W^c_{ijt} \tag{A.3}$$

From the batchsize constraints (3.27), summing over over time t we get:

$$V_{ij}^{\max} \sum_{t=1}^{T_c} W_{ijt}^{c} \geq \sum_{t=1}^{T_c} B_{ijt}^{c} = B_{ij}^{c*} \tag{A.4}$$

where $V_{ij}^{\max} = \phi_{ij}^{\max} V_j$.

Combining (A.3) and (A.4), we get:

$$V_{ij}^{\max} \left\lfloor \frac{T_c}{p_i} \right\rfloor \geq B_{ij}^{c*} \geq 0 \tag{A.5}$$

We now consider (A.2) and (A.5) together. It is obvious that the ΔS_{sc} variable will take its maximum value when all the B_{ij}^{c*} in the first summation of (A.2) are set to their maximum value while the ones in the second summation are set to zero. In any case, ΔS_{sc} cannot exceed the available storage capacity C_s. We therefore have:

$$\Delta S_{sc}^{\max} = \min \left(C_s, \sum_{i \in \bar{T}_s} \bar{\rho}_{is} \left\lfloor \frac{T_c}{p_i} \right\rfloor \sum_{j \in K_i} V_{ij}^{\max} \right) \tag{A.6}$$

Similarly, the minimum value of ΔS_{sc} is taken when the B_{ijt}^{c*} in the first summation are set to zero while those in the second one take their maximum value:

$$\Delta S_{sc}^{\min} = -\min \left(C_s, \sum_{i \in T_s} \rho_{is} \left\lfloor \frac{T_c}{p_i} \right\rfloor \sum_{j \in K_i} V_{ij}^{\max} \right) \tag{A.7}$$

It should be noted that $\Delta S_{sc}^{\min}$ and $\Delta S_{sc}^{\max}$ may be calculated *a priori* for every state s in every campaign c through preprocessing procedures, thus resulting in much tighter form of (3.12a, b). In particular, for feed states, $\bar{T}_s = \{\} \rightarrow \Delta S_{sc}^{\max} = 0$. Similarly, for products states, $T_s = \{\} \rightarrow \Delta S_{sc}^{\min} = 0$. Obviously, one side of (3.12a, b) may be omitted from the formulation for such states.

A.2 MILP-based Bounds Derivation

One problem with the above bounds is that they are derived by considering each state in isolation, taking account only of the processing capacity available to the tasks with which it interacts directly. However, very often the nett change in the inventory of state is restricted not only by limited processing capacity but also by the limited availability of other types of material used for producing this state. Further improvement of these bounds may be obtained by solving a series of MILPs for every state s in each campaign c:

$$\max_{(s,\,c)} \left(S^c_{s,T_c+1} - S^c_{s0}\right)$$

$$s.t.\ \ \textit{Cyclic Scheduling Constraints}$$

This procedure may sometimes require large computational times if a large number of MILPs needs to be solved. There are two approaches to overcoming this difficulty:

Either

solve only the LP relaxation of the above problem;

or solve the MILP but replacing the exact cyclic scheduling constraints by their aggregate versions (see, for instance, constraints (5.1)-(5.7)).

Appendix B

Generation of Valid Inequalities

Preprocessing algorithms for obtaining valid inequalities from constraints of the form:

$$\sum_j a_{ij} x_j \geq b_i \quad \forall\, i \tag{B.1}$$

are considered. Our algorithms employ redundancy checks to ensure generation of a minimal set of constraints; they also exploit the sparsity of the constraints. All the constraints generated are of pure integer form.

We describe two preprocessing algorithms to obtain valid inequalities for pure integer constraints. For reasons of simplicity, we consider a single constraint of the form:

$$\sum_{j \in N} a_j x_j \geq b \tag{B.2}$$

$$x_j \in \{0, 1\}$$

$$a \in R_+^N$$

$$b \in R_+$$

where N is a set of indices, $N = \{1, 2, \ldots, n\}$.

B.1 Coefficient Reduction

This procedure, implemented in the IBM MPSX package (1990), is useful when there are coefficients, a_j, greater than the right hand side, b (B.2). In this case, they can be reduced to the value of b so as to avoid fractional solutions.

Constraint (B.2) can be written as:

$$\sum_{j \in N} a_j x_j = \sum_{j \in N-\{k\}} a_j x_j + a_k x_k \geq \sum_{j \in N-\{k\}} a_j x_j + b x_k \tag{B.3}$$

where $a_k \geq b$.

From constraint (B.3), we can derive the following inequality to replace constraint (B.2):

$$\sum_{j \in N - \{k\}} a_j x_j + b x_k \geq b \tag{B.4}$$

Note that the feasible region defined by (B.4) is the same as that for (B.2). Thus, if $x_k = 1$, then (B.4) is valid as a consequence of non-negativity of a_j, and x_j. If $x_k = 0$, then the summation of (B.4) must be greater than b, otherwise (B.2) would be violated too.

The new constraint (B.4) is tighter than (B.2) as it has smaller coefficients and the same right hand side.

The procedure can be applied iteratively for all the coefficients which are greater than b using the rule:

IF $a_k > b$, THEN $a_k := b$.

For example, consider the constraint $8X_1 + 7X_2 + 3X_3 + 2X_4 \geq 4.0$. The new constraint will be $4X_1 + 4X_2 + 3X_3 + 2X_4 \geq 4.0$.

B.2 Cut Constraints

In this section, we describe a method for deriving valid inequalities from constraints of the form (B.2). The procedure is similar to that described by Williams (1990), but is extended to ensure that no redundant constraints are included in the final set.

The coefficients of constraint (B.2) are sorted in descending order. Let X be the feasible region of constraint (B.2). We define a *cut* as a pair (J, m) of a set $J \subseteq N$ and a cardinal $m \in [1, n]$ for which the constraint:

$$\sum_{j \in J} x_j \geq m \quad \forall\ x \in X \tag{B.5}$$

is always true within the feasible region of (B.2).

Let j_l, $l = 1..e(J)$ be the index of the lth element of J where $e(J)$ is the cardinality of J, the latter being ordered so that $a_{j_l} \geq a_{j_{l+1}}$, $l = 1..e(J) - 1$.

Proposition: (J, m) is a cut if and only if

$$\sum_{l=m}^{e(J)} a_{j_l} > \left(-b + \sum_{j \in N} a_j \right) \tag{B.6}$$

Proof: Note that the left hand side of (B.6) is the sum of the smallest $e(J) - m + 1$ elements of

J.

IF: Rearranging (B.6) we obtain:

$$\sum_{j \in N} a_j - \sum_{l=m}^{e(J)} a_{j_l} < b \tag{B.7}$$

or

$$\sum_{l=1}^{m-1} a_{j_l} + \sum_{j \notin J} a_j < b \tag{B.8}$$

Taking into consideration the ordered nature of J we have:

$$\sum_{l=1}^{m-1} a_{j_l} \geq \sum_{j \in \tilde{J}_{m-1}} a_j \quad \forall\ \tilde{J}_{m-1} \subseteq J : e(\tilde{J}) = m-1 \tag{B.9}$$

$$B.8 \overset{B.9}{\rightarrow} \sum_{j \in \tilde{J}_{m-1}} a_j + \sum_{j \notin J} a_j < b \tag{B.10}$$

This implies that, were all but $m-1$, variables in set J to be set to zero, it would be impossible to satisfy (B.2) even if all other variables were set to one.

Thus at least m variables in J must take a value of one, which is expressed mathematically through the constraint (B.5).

ONLY IF: Assume that (B.6) is not satisfied, *i.e.*

$$\sum_{l=m}^{e(J)} a_{j_l} \leq -b + \sum_{j \in N} a_j \tag{B.11}$$

or

$$\sum_{j \in N} a_j - \sum_{l=m}^{e(J)} a_{j_l} \geq b \tag{B.11$'$}$$

or

$$\sum_{j \notin J} a_j + \sum_{l=1}^{m-1} a_{j_l} \geq b \tag{B.11$''$}$$

Then the vector, $\underline{x}$:

$$x_j = 1 \quad \forall\ j \notin J$$

$$x_{j_l} = 1, \quad l = 1..m-1$$

$$x_{j_l} = 0, \quad l = m..e(J)$$

(a) belongs to the feasible region X because of (B.11), but

(b) does not satisfy (B.5) ($\sum_{j \in J} x_j = m - 1$)

hence (J, m) is not a cut. **Q.E.D.**

By checking condition (B.6), we can easily construct all cuts (J, m) for a given constraint (B.2). However, some of these may be redundant. Below, we examine various ways in which this redundancy can arise, and how it may be detected and avoided.

Redundancy Type I: (J, m) is redundant with respect to (J, m') if $m' \geq m$.

This is an obvious consequence of (B.5). It can simply be avoided by selecting the largest m which satisfies (B.6) for a given set J. For example, $X_1 + X_2 + X_3 \geq 1$ is redundant with respect to $X_1 + X_2 + X_3 \geq 2$.

Redundancy Type II: (J, m) is redundant with respect to (J', m) if $J' \subseteq J$.

This is a consequence of (B.11) and the non-negativity of x_j. For example, $X_1 + X_2 + X_3 + X_4 \geq 2$ is redundant with respect to $X_1 + X_2 + X_3 \geq 2$.

To detect this, once we have identified a cut (J, m), we consider dropping the *ith* element of J to form J'. There are two cases to consider:

(i) $i < m$

element dropped

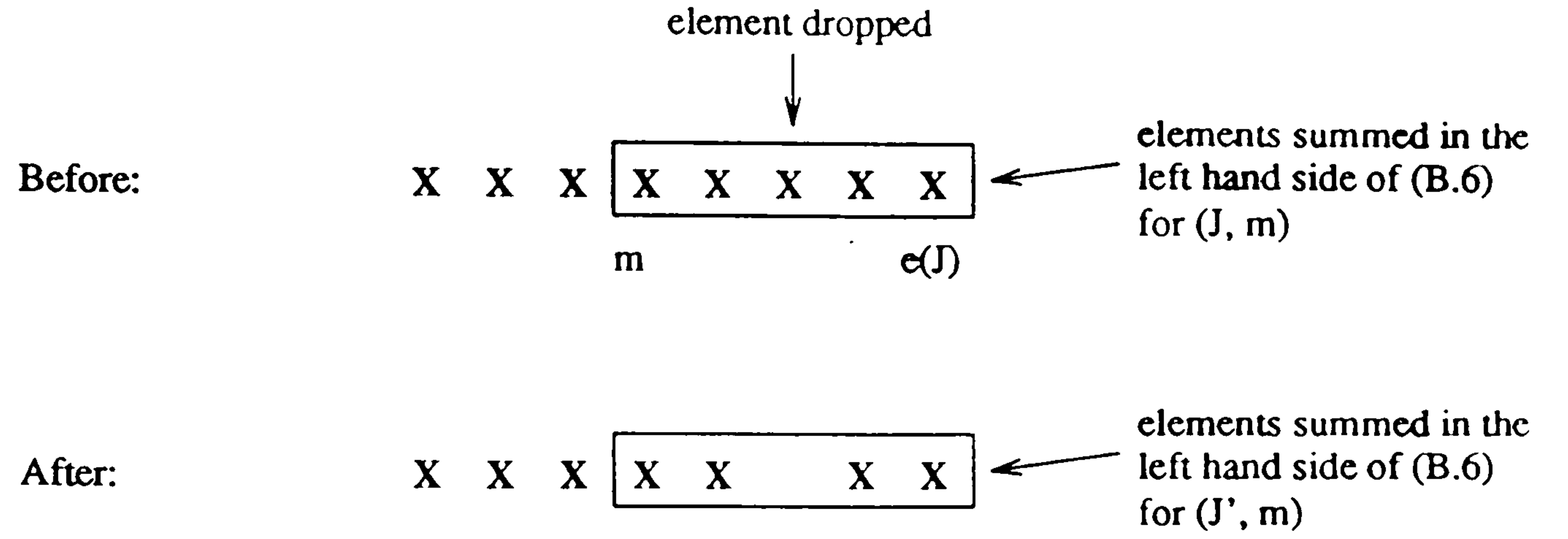
Before: X X X $\boxed{\text{X X X X X}}$ ← elements summed in the left hand side of (B.6) for (J, m)

m e(J)

After: X X X $\boxed{\text{X X X X}}$ ← elements summed in the left hand side of (B.6) for (J', m)

We see that the sum on the left hand side of (B.6) for (J', m) is the same as that for the left hand side of (J, m) minus a_{j_m}.

(ii) $i \geq m$



Now the sum on the left hand side is equal to the original sum minus a_{j_i}.

Thus, in both cases, (NewSum) = (OldSum) $- a_{j_{max(i,\,m)}}$.

For (J, m) to be redundant, we want the new sum to exceed the right hand side of (B.6). Thus, to test for redundancy of this type we should select i to be as large as possible, *i.e.* $i = e(J)$.

$$\sum_{l=m}^{e(J)-1} a_{j_l} > -b + \sum_{j \in N} a_j \tag{B.12}$$

Redundancy Type III: (J, m) is redundant with respect to (J', m') if

(i) $J \subset J'$, and

(ii) $m' - m \geq e(J') - e(J)$.

This is a consequence of (B.6) and the fact that $x_j \leq 1$, since (J', m') is a cut, we have:

$$\sum_{j \in J'} x_j \geq m' \;\rightarrow\; \sum_{j \in J} x_j + \sum_{j \in J' \setminus J} x_j \geq m' \;\rightarrow$$

$$\sum_{j \in J} x_j \geq m' - \sum_{j \in J' \setminus J} x_j \geq m' - [e(J') - e(J)] \tag{B.13}$$

which, if (ii) holds, implies that $\sum_{j \in J} x_j \geq m$, which is the (J, m) cut.

For example, $X_1 + X_2 \geq 1$ is redundant with respect to $X_1 + X_2 + X_3 \geq 2$.

To detect this type of redundancy, we consider if it is possible to generate a cut $(J + \{i\}, m + 1)$ from (J, m) by adding an element $i \notin J$ to J.

Once again, we consider two cases:

(i) $a_i \geq a_{j_m}$: the new element is thus inserted before the *mth* element of J.

```
Before:    X X     X X |X X X X X X|
                        m         e(J)

After:     X X X X X |X X X X X X|
               ↑      m+1        e(J')
          new element
```

We see that the sum on the left hand side of (B.6) actually involves the same elements before and after. Hence, (B.6) is satisfied and (J, m) is redundant.

(ii) $a_i < a_{j_m}$: the new element is inserted after the *mth* element of J.

```
Before:    X X X X X |X X     X X X|
                      m         e(J)

After:     X X X X X X |X X X X X|
                    m+1  ↑      e(J')
                    new element
```

We see that the sum involved is the original one minus a_{j_m} plus a_i. If this satisfies (B.6), then (J, m) is redundant. (*NB*. NewSum < OldSum since $a_i < a_{j_m}$).

In any case, to show that (J, m) is redundant, we want the new sum to be as large as possible — thus we select:

$$a_i = \max_{j \notin J}(a_j) \tag{B.14}$$

If $a_i \geq a_{j_m}$ or $\sum_{l=m+1}^{e(J)} a_j + a_i > \left(-b + \sum_{j \in N} a_j\right)$, then (J, m) is redundant. This can be written as a single check:

$$\sum_{l=m+1}^{e(J)} a_j + \max_{j \notin J}(a_j) > \left(-b + \sum_{j \in N} a_j\right) \tag{B.15}$$

Redundancy Type IV: (J, m) is redundant with respect to (J', m') and (J'', m'') if

(i) $J' \subset J$,

(ii) $J'' \subset J$, and

(iii) $m' + m'' - e(J' \cap J'') \geq m$.

Since (J', m') and (J'', m'') are cuts, we have:

$\sum_{j \in J'} x_j \geq m'$ and $\sum_{j \in J''} x_j \geq m''$.

Adding these two together, we get:

$$\sum_{j \in J' \cup J''} x_j + \sum_{j \in J' \cap J''} x_j \geq m' + m'' \tag{B.16}$$

Now, since $J', J'' \subset J$, we also have $J' \cup J'' \subset J$, which implies that

$$\sum_{j \in J} x_j \geq \sum_{j \in J' \cup J''} x_j \geq m' + m'' - \sum_{j \in J' \cap J''} x_j \geq m' + m'' - e(J' \cap J'') \geq m \tag{B.17}$$

The last inequality is true because of condition (iii). Therefore, (J, m) is implied by the "smaller" cuts (J', m') and (J'', m'').

For example, $X_1 + X_2 + X_3 + X_4 \geq 2$ is redundant with respect to $X_1 \geq 1$ and $X_3 + X_4 \geq 1$.

Note that the extension of the above algorithm of single constraint to the case of multiple constraints is straightforward. Thus, we use a recursive procedure for generating cuts for each constraint in sequence. However, during the redundancy checks, we take into consideration not only the cuts of the constraint currently being processed, but also those generated from earlier constraints.

B.3 Example

We consider the following 8 constraints.

$$8X_1 + 4X_2 + 3X_3 + 2X_4 \geq 3.5$$

$$7X_1 + 3X_2 + 7X_4 + X_5 \geq 10.0$$

$$4X_1 + X_2 + 4X_3 + X_4 \geq 7.0$$

$$X_1 + 4X_2 + 2X_3 + 4X_4 + 2X_5 \geq 8.0$$

$$4X_1 + 3X_2 + 2X_3 \geq 1.0$$

$$3X_1 + 2X_2 + X_3 + X_4 \geq 4.0$$

$$X_2 + 3X_3 + 2X_4 + 2X_5 + X_6 \geq 6.0$$

$$X_1 + 2X_2 + X_3 + 4X_4 + 5X_5 \geq 7.0$$

Applying the cut generation algorithm for each constraint separately, we obtain:

Constraint 1	$X_1 + X_2 + X_4 \geq 1$
	$X_1 + X_2 + X_3 \geq 1$
Constraint 2	$X_1 + X_2 + X_4 \geq 2$
Constraint 3	$X_1 + X_3 \geq 2$
Constraint 4	$X_2 + X_4 + X_5 \geq 2$
	$X_2 + X_3 + X_4 \geq 2$
Constraint 5	$X_1 + X_2 + X_3 \geq 1$
Constraint 6	$X_1 + X_3 \geq 1$
	$X_1 + X_4 \geq 1$
	$X_1 + X_2 \geq 1$
	$X_1 + X_2 + X_3 + X_4 \geq 2$
Constraint 7	$X_3 + X_6 \geq 1$
	$X_2 + X_3 \geq 1$
	$X_3 + X_4 + X_5 \geq 1$
	$X_2 + X_3 + X_4 + X_5 + X_6 \geq 3$
Constraint 8	$X_1 + X_3 + X_5 \geq 1$
	$X_2 + X_5 \geq 1$
	$X_4 + X_5 \geq 1$
	$X_1 + X_2 + X_4 + X_5 \geq 2$
	$X_2 + X_3 + X_4 + X_5 \geq 2$

Considering all the constraints together, we obtain a much smaller set:

$$\boxed{\begin{array}{c} X_1 + X_2 + X_4 \geq 2 \\ X_1 + X_3 \geq 2 \\ X_2 + X_4 + X_5 \geq 2 \\ X_2 + X_3 + X_4 \geq 2 \\ X_3 + X_6 \geq 1 \\ X_3 + X_4 + X_5 \geq 2 \\ X_1 + X_3 + X_5 \geq 1 \end{array}}$$

It can be verified that these dominate all the cuts derived from individual constraints. In fact the set could be reduced further by taking into account that $X_1 = X_3 = 1$, but this has not been implemented.

Appendix C

Derivation of Bounds on HTM Flowrates and Exit Temperatures

We start by solving the energy balance equations (6.5) and (6.6) for the HTM flowrate f_{ij} and the temperature Θ_{ij}^{out} of the HTM leaving the heat-integrated operation:

$$f_{ij} = \frac{1}{2c_p\left(\frac{T_{ij} - \Theta_{s_{ij}}}{q_{ij}} - \frac{1}{U_{ij}A_{ij}}\right)} \tag{C.1}$$

and

$$\Theta_{ij}^{out} = 2\left(T_{ij} - \frac{q_{ij}}{U_{ij}A_{ij}}\right) - \Theta_{s_{ij}} \tag{C.2}$$

We distinguish two cases corresponding to exothermic ($q_{ij} > 0$) and endothermic ($q_{ij} < 0$) heat-integrated operations.

Case I: Exothermic Heat-Integrated Operations

In this case, $q_{ij} > 0$ and $T_{ij} > \Theta_{s_{ij}}$. It can be seen from (C.1) that f_{ij} is maximized if the quantity $(T_{ij} - \Theta_{s_{ij}})/q_{ij}$ is minimized. This is achieved when $\Theta_{s_{ij}}$ is at its upper bound. Since the other two variables, T_{ij} and q_{ij} vary over the duration of the task, we must select the time at which this quantity is minimized.

Furthermore we note that the heat load $q_{ij}(t)$ is, in fact, a non-decreasing function of batchsize, and in order to minimize $(T_{ij} - \Theta_{s_{ij}})/q_{ij}$ we have to consider the heat load for the largest possible batchsize of task i taking place in unit j. This is denoted by $V_{ij}^{\max}$.

The above reasoning, together with the pumping rate constraints (6.16), lead to the bound

$$f_{ij}^{\max} = \min\left[\frac{1}{2c_p\left(\min\limits_{t\in[0,\,p_i]} \frac{T_{ij}(t) - \Theta_{s_{ij}}^{\max}}{q_{ij}(t, V_{ij}^{\max})} - \frac{1}{U_{ij}A_{ij}}\right)},\ F_j^{\max}\right] \tag{C.3}$$

From (C.2) we see that in order to maximize Θ_{ij}^{out}, $\Theta_{s_{ij}}$ must attain its minimum value and the quantity $(T_{ij} - q_{ij}/U_{ij}A_{ij})$ its maximum. Furthermore, we must consider the batchsize that minimizes q_{ij}, *i.e.* the minimum batchsize for task i taking place in unit j, denoted by $V_{ij}^{\min}$. We therefore have:

$$\Theta_{ij}^{out,max} = 2 \max_{t \in [0,\, p_i]} \left(T_{ij}(t) - \frac{q_{ij}(t, V_{ij}^{\min})}{U_{ij}A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.4a}$$

With similar reasoning, we obtain the following lower bound on Θ_{ij}^{out}:

$$\Theta_{ij}^{out,min} = 2 \min_{t \in [0,\, p_i]} \left(T_{ij}(t) - \frac{q_{ij}(t, V_{ij}^{\max})}{U_{ij}A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.4b}$$

The bounds (C.3) and(C.4) are expressed in the continuous time domain. We now turn to the issue of establishing analogous bounds on the discrete variables $fijt$ and Θ_{ijt}^{out} which correspond to the values of $f_{ij}(t)$ and $T_{ij}(t)$ at the start of discrete intervals. Taking account of the fact that the heat load q_{ij} is a piecewise constant function of time defined by (6.9), whilst the temperature T_{ij} is continuous at the interval boundaries, we arrive at the expressions:

$$f_{ij}^{\max} = \min \left[\frac{1}{2c_p \left(\min\limits_{\theta \in [0,\, p_i - 1]} \dfrac{\overline{T}_{ij\theta} - \Theta_{s_{ij}}^{\max}}{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}} - \dfrac{1}{U_{ij}A_{ij}} \right)},\ F_j^{\max} \right] \tag{C.5}$$

$$\Theta_{ij}^{out,max} = 2 \max_{\theta \in [0,\, p_i - 1]} \left(\overline{T}_{ij\theta} - \frac{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\min}}{U_{ij}A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.6a}$$

$$\Theta_{ij}^{out,min} = 2 \min_{\theta \in [0,\, p_i - 1]} \left(\overline{T}_{ij\theta} - \frac{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}}{U_{ij}A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.6b}$$

The bounds on the values of $f_{ij}(t)$ and $T_{ij}(t)$ at the *end* of each discrete interval are similar to (C.5) and (C.6) with the difference that $\overline{T}_{ij\theta}$ must be replaced by $\overline{T}_{ij,\theta+1}$:

$$f_{ij}'^{\max} = \min \left[\frac{1}{2c_p \left(\min\limits_{\theta \in [0,\, p_i - 1]} \dfrac{\overline{T}_{ij,\theta+1} - \Theta_{s_{ij}}^{\max}}{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}} - \dfrac{1}{U_{ij}A_{ij}} \right)},\ F_j^{\max} \right] \tag{C.7}$$

$$\Theta_{ij}^{\prime out,max} = 2 \max_{\theta \in [0,\, p_i - 1]} \left(\overline{T}_{ij,\theta+1} - \frac{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\min}}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.8a}$$

$$\Theta_{ij}^{\prime out,min} = 2 \min_{\theta \in [0,\, p_i - 1]} \left(\overline{T}_{ij,\theta+1} - \frac{\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.8b}$$

Case II: Endothermic Heat-Integrated Operations

In this case $q_{ij} < 0$ and $T_{ij} < \Theta_{s_{ij}}$. To facilitate the derivation of the bounds, we write (C.1) and (C.2) in the form:

$$f_{ij} = \frac{1}{2c_p \left(\frac{\Theta_{s_{ij}} - T_{ij}}{-q_{ij}} - \frac{1}{U_{ij} A_{ij}} \right)} \tag{C.1$'$}$$

and

$$\Theta_{ij}^{out} = 2\left(T_{ij} + \frac{-q_{ij}}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}} \tag{C.2$'$}$$

We can now see from (C.1′) that f_{ij} is maximized if the quantity $(\Theta_{s_{ij}} - T_{ij})/(-q_{ij})$ is minimized. This is achieved when $\Theta_{s_{ij}}$ is at its lower bound and the batchsize is set at its maximum value. This leads to:

$$f_{ij}^{\max} = \min \left[\frac{1}{2c_p \left(\min_{t \in [0,\, p_i]} \frac{\Theta_{s_{ij}}^{\min} - T_{ij}(t)}{-q_{ij}(t, V_{ij}^{\max})} - \frac{1}{U_{ij} A_{ij}} \right)} ,\, F_j^{\max} \right] \tag{C.9}$$

From (C.2′) we see that in order to maximize Θ_{ij}^{out}, $\Theta_{s_{ij}}$ must attain its minimum value and $-q_{ij}$ its maximum. We therefore have:

$$\Theta_{ij}^{out,max} = 2 \max_{t \in [0,\, p_i]} \left(T_{ij}(t) + \frac{-q_{ij}(t, V_{ij}^{\max})}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.10a}$$

On the other hand, Θ_{ij}^{out} is minimized by setting $\Theta_{s_{ij}}$ to its maximum value and the batchsize to its minimum:

$$\Theta_{ij}^{out,min} = 2 \min_{t \in [0,\, p_i]} \left(T_{ij}(t) - \frac{q_{ij}(t, V_{ij}^{\min})}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.10b}$$

Just as for case I, we now proceed to derive the following bounds on the variables in the discrete time domain:

$$f_{ij}^{\max} = \min \left[\frac{1}{2c_p \left(\min_{\theta \in [0,\, p_i - 1]} \frac{\Theta_{s_{ij}}^{\min} - \bar{T}_{ij\theta}}{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}|} - \frac{1}{U_{ij} A_{ij}} \right)},\, F_j^{\max} \right] \tag{C.11}$$

$$\Theta_{ij}^{out,max} = 2 \max_{\theta \in [0,\, p_i - 1]} \left(\bar{T}_{ij\theta} + \frac{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}|}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.12a}$$

$$\Theta_{ij}^{out,min} = 2 \min_{\theta \in [0,\, p_i - 1]} \left(\bar{T}_{ij\theta} + \frac{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\min}|}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.12b}$$

$$f_{ij}'^{\max} = \min \left[\frac{1}{2c_p \left(\min_{\theta \in [0,\, p_i - 1]} \frac{\Theta_{s_{ij}}^{\min} - \bar{T}_{ij,\theta+1}}{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}|} - \frac{1}{U_{ij} A_{ij}} \right)},\, F_j^{\max} \right] \tag{C.13}$$

$$\Theta_{ij}'^{out,max} = 2 \max_{\theta \in [0,\, p_i - 1]} \left(\bar{T}_{ij,\theta+1} + \frac{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}|}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\min} \tag{C.14a}$$

$$\Theta_{ij}'^{out,min} = 2 \min_{\theta \in [0,\, p_i - 1]} \left(\bar{T}_{ij,\theta+1} + \frac{|\gamma_{ij\theta}^{(1)} + \gamma_{ij\theta}^{(2)} V_{ij}^{\max}|}{U_{ij} A_{ij}} \right) - \Theta_{s_{ij}}^{\max} \tag{C.14b}$$

The bounds (C.5)-(C.8) and (C.11)-(C.14) are those used in the linearizations (6.18) of the bilinear terms defined by (6.17).

Appendix D

Batch Unit Operation Models

Here we present the dynamic models used for the optimal control computations of chapter 7.

D.1 Batch Reactor Model

Assumptions:

- Well-stirred reactor.

Notation:

N_k	:	Molar holdup of component k.
r_i	:	Reaction rate of reaction i.
v_{ik}	:	Stoichiometriccoefficient of component k in reaction i.
V	:	Volume.
H_R	:	Liquid enthalpy.
h_k^l	:	Molar liquid enthalpy of component k.
Q_{cR}	:	Energy removed by external cooling system.
Q_x	:	Energy exchanged.
ΔH_{Ri}	:	Heat of reaction for reaction i.
K_i	:	Rate constant of reaction i.
C_k	:	Concentration of component k.
β_{ki}	:	Reaction order of component k in reaction i.
ρ_k	:	Molar density of component k.
T_R	:	Temperature.
A_i	:	Frequency for reaction i.
E_i	:	Activation energy for reaction i.
R	:	Universal gas constant.
t	:	Time.

Reactor

Component Mass Balance:

$$\frac{dN_k}{dt} = V \sum_i r_i \, v_{ik} \quad \forall\, k \tag{D.1}$$

Energy Balance:

$$\frac{dH_R}{dt} = V \sum_i r_i \, (-\Delta H_{Ri}) - Q_{cR} - Q_x \tag{D.2}$$

Other Equations:

$$r_i = K_i \prod_k C_k^{\beta_{ki}} \quad \forall\, i \tag{D.3}$$

$$C_k = \frac{N_k}{V} \quad \forall\, k \tag{D.4}$$

$$V = \sum_k \frac{N_k}{\rho_k} \tag{D.5}$$

$$K_i = A_i \, e^{-E_i/RT_R} \quad \forall\, i \tag{D.6}$$

$$H_R = \sum_k N_k \, h_{Rk}^l \tag{D.7}$$

$$h_k^l = h(T_R) \quad \forall\, k \tag{D.8}$$

D.2 Batch Distillation Column Model

Assumptions:

- no reaction,
- negligible vapour holdup,
- adiabatic tray operation,
- perfect mixing,
- constant plate and condenser holdups,
- negligible column hydraulics, and

- fast energy dynamics.

The distillation column is split in four sections, namely: collector, total condenser, trays and reboiler.

Notation[1]:

x_{sk}	:	Liquid mole fraction of component k in column section s.
y_{sk}	:	Vapour mole fraction of component k in column section s.
M_s	:	Total holdup in column section s.
V_s	:	Total vapour flow leaving section s.
L_s	:	Total distillate flow from section s.
L_D	:	Total liquid flow leaving the column.
r	:	Reflux ratio.
T_s	:	Temperature of tray in column section s.
K_{sk}	:	Vapour equillibrium constant of component k in column section s.
Q_s	:	Energy lost/supplied to section s by external heating system.
Q_x	:	Energy exchanged at reboiler section.
h_s^l/h_s^v	:	Molar liquid/vapour enthalpy of section s.
t	:	Time.

Accumulator

Accumulator Mass Balance:

$$\frac{dM_A}{dt} = L_D \tag{D.9}$$

Component Mass Balance:

$$M_A \frac{dx_{Ak}}{dt} = L_D (x_{Ck} - x_{Ak}) \ \forall \ k \tag{D.10}$$

Total Condenser

Component Mass Balance:

$$M_C \frac{dx_{Ck}}{dt} = V_1 \, y_{1k} - x_{Ck} (L_D + L_C) \ \forall \ k \tag{D.11}$$

---

[1] Here s is one of n, C, A, and R in order to describe the trays, the condenser, accumulator and reboiler respectively.

Energy Balance:

$$Q_C = V_1\ (h_1^v - h_C^l) \tag{D.12}$$

Bubble Point Calculation:

$$\sum_k K_{Ck}\ x_{Ck} = 1 \tag{D.13}$$

Other Equations:

$$V_1 = L_C + L_D \tag{D.14}$$

$$L_C = r\ V_1 \tag{D.15}$$

Trays

Total Mass Balance:

$$L_{n-1} + V_{n+1} = L_n + V_n \quad \forall\ n \tag{D.16}$$

Component Mass Balance:

$$M_n\ \frac{dx_{nk}}{dt} = L_{n-1}\ x_{n-1,k} - L_n\ x_{nk} + V_{n+1}\ y_{n+1,k} - V_n\ y_{nk} \quad \forall\ n,k \tag{D.17}$$

Energy Balance:

$$L_{n-1}\ h_{n-1}^l - L_n\ h_n^l + V_{n+1}\ h_{n+1}^v - V_n\ h_n^v = 0 \quad \forall\ n \tag{D.18}$$

Equilibrium:

$$y_{nk} = K_{nk}\ x_{nk} \quad \forall\ n,k \tag{D.19}$$

Vapour Mole Fraction Normalisation:

$$\sum_k y_{n,k} = 1 \quad \forall\ n \tag{D.20}$$

Reboiler

Overall Mass Balance:

$$\frac{dM_R}{dt} = L_N - V_R \tag{D.21}$$

Component Mass Balance:

$$M_R\ \frac{dx_{Rk}}{dt} = L_N\ (x_{Nk} - x_{Rk}) - V_R\ (y_{Rk} - x_{Rk}) \quad \forall\ k \tag{D.22}$$

Energy Balance:

$$L_N\ (h^l_N - h^l_R) + V_R\ (h^l_R - h^v_R) + Q_R + Q_x = 0 \tag{D.23}$$

Equilibrium:

$$y_{Rk} = K_{Rk}\ x_{Rk} \quad \forall\ k \tag{D.24}$$

Other Equations:

$$\sum_k y_{R,k} = 1 \tag{D.25}$$

Physical Property Calculations

$$K_{sk} = K_k(\ T_s) \quad \forall\ s \tag{D.26}$$

$$h^l_s = h^l(\ x_{sk},\ T_s) \quad \forall\ s \tag{D.27}$$

$$h^v_s = h^v(\ y_{sk},\ T_s) \quad \forall\ s,\ s \neq C \tag{D.28}$$

D.3 Heat Exchange Model

Assumptions:

- well-stirred heat exchanger, and
- fast energy dynamics.

Notation:

Q_x : Energy exchanged.
F_x : Heat transfer medium flowrate.
C_p : Heat transfer medium heat capacity.
U : Overall heat transfer coefficient.
A : Heat exchanger area.
T_r : Reactor temperature.
T_d : Reboiler temperature.
T_{x1} : Heat transfer medium temperature after reactor.
T_{x2} : Heat transfer medium temperature after reboiler.

Heat Exchanger

Heat Exchange Equations:

$$Q_x = F_x\, C_p\, (T_{x1} - T_{x2}) \tag{D.29}$$

$$Q_x = U\ A\ (T_r - T_{x1}) \tag{D.30}$$

$$Q_x = U\ A\ (T_{x2} - T_d) \tag{D.31}$$

D.4 Parameter Values Used in Examples

	Example 1	Example 2
Reaction		
A_1 $(m^3/mole.s)$	0.0033	0.0006
E_1 $(J/mole)$	30000.0	29000.0
ΔH_{R1} $(J/mole)$	-100000.0	-100000.0
ρ_A $(mole/m^3)$	11000.0	11000.0
ρ_B $(mole/m^3)$	16000.0	16000.0
ρ_C $(mole/m^3)$	10400.0	
ρ_D $(mole/m^3)$	10000.0	
h_A^l $(J/mole)$	$172.3(T_R - 298.0)$	$200.0(T_R - 298.0)$
h_B^l $(J/mole)$	$200.0(T_R - 298.0)$	$300.0(T_R - 298.0)$
h_C^l $(J/mole)$	$160.0(T_R - 298.0)$	
h_D^l $(J/mole)$	$155.0(T_R - 298.0)$	
Distillation*		
M_s $(mole)$	10.0	10.0
K_{s1}	$e^{(-4543.71/T_s+11.2599)}/1.025$	$e^{(-4543.71/T_s+11.2599)}/1.025$
K_{s2}	$2.0\,K_{s1}$	$2.0\,K_{s1}$
h_s^l $(J/mole)$	$172.3x_{s1}(T_s - 298.0) + 200.3x_{s2}(T_s - 298.0)$	$200.0x_{s1}(T_s - 298.0) + 300.0x_{s2}(T_s - 298.0)$
h_s^v $(J/mole)$	$y_{s1}(172.3(T_s - 298.0) + 31000.0) + y_{s2}(200.3(T_s - 298.0) + 26000.0)$	$y_{s1}(200.0(T_s - 298.0) + 31000.0) + y_{s2}(300.0(T_s - 298.0) + 26000.0)$
Heat Exchange		
C_p $(J/kg.K)$	2100.0	

*: In Example 2, $s1$ and $s2$ refer to A and B respectively.