7

Application of Distributed Coordination Approach – A Case Example

7.1 Introduction

To illustrate the application of distributed coordination approach, we now describe an example of an industrial-scale, multipurpose process plant. The example is derived from a similar example in Friedler *et al.* (1992) and reflects largely the characteristics of modern process plants in petrochemicals, polymers and chemicals industries, except that, for simplicity, we omit the complexities of recycles or byproducts. These limitations however do not impede the generality of discussions in this chapter. The multipurpose nature of the example allows us to analyse a number of potential production scenarios that can be expected to arise in this class of industry in future. In this sense the example also reflects the long-term vision of a highly reconfigurable process control system and shows that it can be developed using a distributed approach. The system has developed in sufficient detail that it might be used as a benchmark problem.

This chapter is structured as follows. The next section describes the example process considered in this chapter. Section 7.3 then introduces the problem formulation in terms of six different production scenarios used for analysis. The subsequent three sections then apply the developments from previous chapters to example process and these scenarios to illustrate how the proposed distributed coordination would operate under these conditions.

7.2 Process Description

The multipurpose process considered as example comprises 18 process units, each capable of performing one or more processing tasks. Fig. 7.1 diagrammatically shows the initial physical layout of the process, which might, for example, be used for polymer, polyester and some petrochemical products. The process is able to produce three products A, B and C. Fig. 7.2 shows

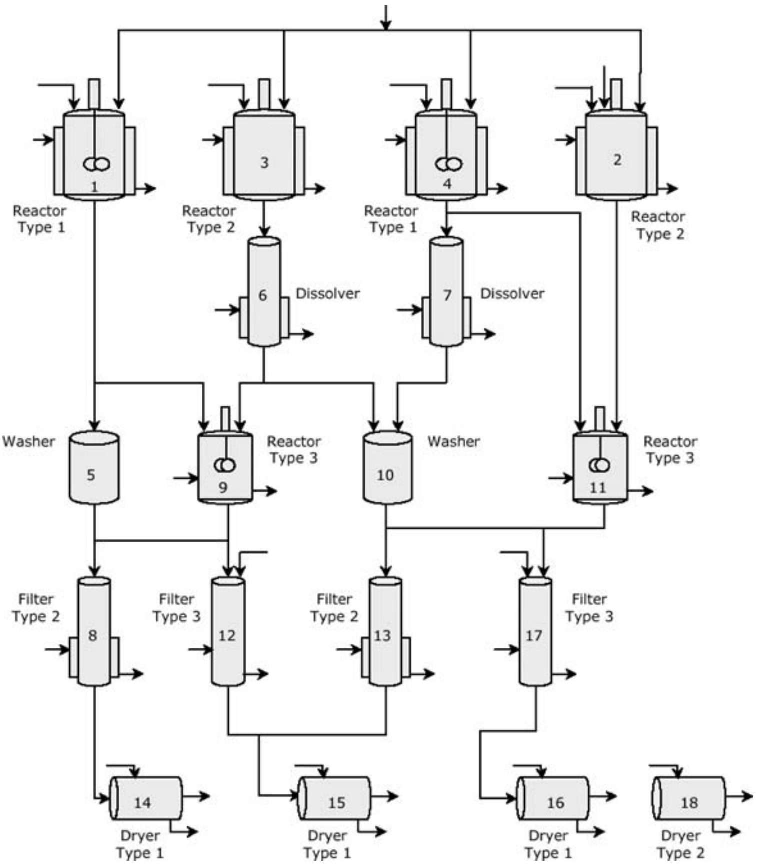


Fig. 7.1. Process layout for multipurpose process example

the initial product recipes for these products in terms of the sequence of processing tasks required to convert raw-materials to end-products. The oblong symbols therein represent the processing tasks while the rectangles represent the materials. The flow of materials is thus from top to bottom. Table 7.1 lists the structure of processing tasks in terms of the associated unit operation and the input and output materials for each task. Note that the recipes for all three products are of *non-linear* type, *i.e.*, there exists more than one task sequence that can produce the same end-product. The dark-lined sequence in each case is the preferred sequence over others.

We note that the tasks in Table 7.1 are not assigned to any processing units at this stage yet. Later in Section 7.4 we consider three different combinations of these 'initial' physical layout and product recipes to understand how the process of managing task assignment, *i.e.*, recipe mapping, works and the various physical and product issues that surround it.

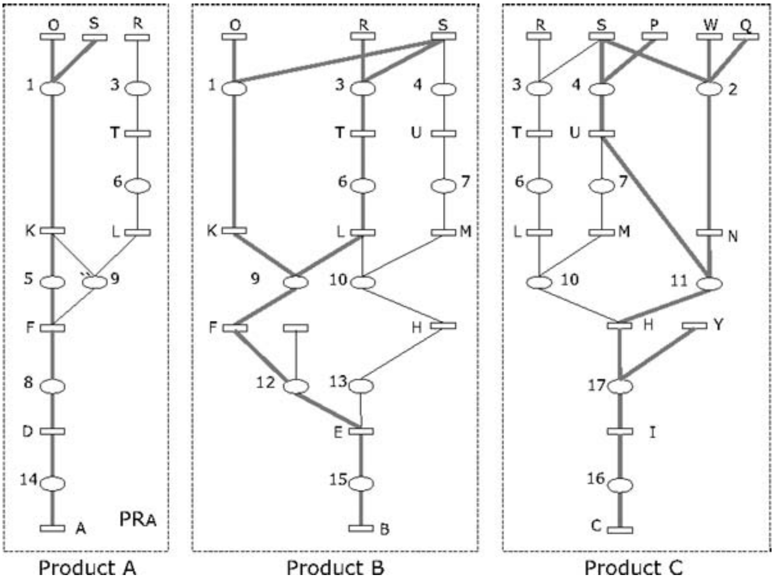


Fig. 7.2. Product recipes for products *A*, *B*, and *C* (the dark lines show the preferred task sequence

As can be seen we have omitted services in both Figs 7.1 and 7.2 in order to simplify the discussions, and to also focus on the key aspect of demand-pull

Table 7.1. List of processing tasks in multipurpose process example

No.	Type	Inputs	Outputs
1.	Reactor	O,S	K
2.	Reactor	S,W,Q	N
3.	Reactor	S,R	T
4.	Reactor	S,P	U
5.	Washer	K	F
6.	Dissolver	T	L
7.	Dissolver	U	M
8.	Filter	F	D
9.	Reactor	K,L	F
10.	Washer	L,M	H
11.	Reactor	U,N	H
12.	Filter	F,G	E
13.	Filter	H	E
14.	Dryer	D	A
15.	Dryer	E	B
16.	Dryer	I	C
17.	Filter	H Y	I
18.	Dryer	X	B

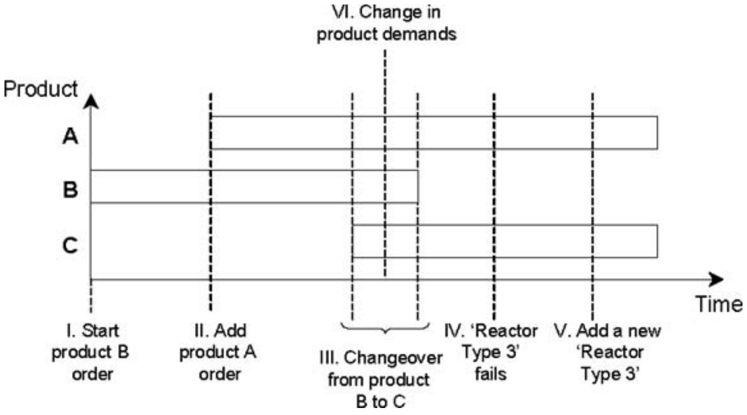


Fig. 7.3. Schedule of product campaigns and example scenarios

type behaviour of process elements. The description in this chapter can simply be extended to also cover service flows.

7.3 Problem Description

Our main purpose for the example considered in this chapter is to analyse the ability of a process control system to respond to a change in plant condition that demands a level of reconfigurability from process operations. To perform this analysis systematically, we propose a set of production scenarios representing the type of changes or disturbances that can be expected to arise in future process operations. These scenarios, while not fully exhaustive, illustrate a number of the features of the proposed distributed approach in a best possible way. Fig. 7.3 shows a time-line for these scenarios within a production run. They are defined as follows.

- **Scenario I – Start product B order:** Assume the process is idle at start, and that a new order for product B arrives. A campaign for producing B is initiated (This scenario should illustrate the process of integrating product recipe information in developing a process scheme);
- **Scenario II – Add product A order:** While B is being produced, assume an order for product A also arrives and is initiated immediately (This scenario should illustrate the process of managing change, in particular for those units which can be involved in both process schemes).
- **Scenario III – Changeover from product B to C:** While the order for B is nearing to its completion, assume an order for C arrives and is initiated in parallel to A and B. On completion, the order for B is stopped and removed (This scenario, similar to scenario II, should depict

Table 7.2. Links between scenarios and reconfigurability requirements

		Scenario					
		I	II	III	IV	V	VI
Requirements	Product/process diversity	✓	✓	✓			
	Easy modifiability		✓	✓	✓		
	Responsiveness		✓	✓	✓		✓
	Fault-tolerance					✓	

the process of changeover but in an opposite way, *i.e.,* the product B is now also removed).

- **Scenario IV – 'Reactor Type 3' fails:** Assume at this stage one of the 'reactor type 3' fails, and is unable to perform its task. (This scenario should illustrate the ability of the system to tackle failures and support graceful degradation of performance);
- **Scenario V – Add a new 'Reactor Type 3':** Finally, assume a new 'reactor type 3' is added in its place or assume the failed unit recovers after repair (This scenario should illustrate the modifiability of the network to support inclusion of a new facility, *e.g.*, a new process unit in this case).

The above scenarios illustrate the qualitative aspects of reconfiguration, which in the distributed case refer to architectural and interaction issues. In addition, a sixth scenario is considered below to demonstrate the quantitative aspect of the system to respond to a control change or disturbance.

- **Scenario VI – Change in product demands:** Assume all three products are being produced at the same time (*e.g.*, during changeover from B to C) and that the demands for all three change by 10 deviation units from their current demand set-points (This scenario should illustrate the responsiveness of the system in terms of propagation of demand changes to whole process network).

The above scenarios directly relate to four RPC requirements provided in Chapter 2 as shown in Table 7.2.

7.4 Application of the DRPC Approach

We now describe the distributed approach applied to this example. The description below follows the outline of developments in the previous three chapters, *i.e.,* (i) identification, (ii) organisation from Chapter 4 and (iii) interaction behaviour of process elements from Chapter 5 and (iv) coordination of their distributed process parameters from Chapter 6.

7.4.1 Identification of Process Elements

The identification of process elements is carried out based on the physical structure of the process.

- *Unit Elements:* Each process unit in Fig. 7.1 is associated with a separate unit element in the control architecture. In total, this results in 18 unit elements, each having a capability to perform one or more processing tasks from Table 7.1. The exact number of task(s) that a unit element performs is varied between a single task or multiple tasks as discussed in the next subsection.
- *Header Elements:* Each piping network connecting unit elements in subsequent stages in Fig. 7.1 is represented by a separate header element. This basically results in a header element for each raw-material, intermediate material and end-product. However, in a more flexible layout, as shown later in Figs. 7.5 and 7.6, the header elements can also be associated with more than one material types. We note that not all piping segments in the process need to be identified as header elements (*e.g.*, the connection between units 3 and 6 in Fig. 7.1) if their role is purely to connect two or more unit elements with no added decisions about process or routing flexibility.
- *Service Elements:* While services are omitted from discussions here, the suppliers of each service used by unit or header elements in Fig. 7.1 can be represented by an appropriate service element.
- *Product Elements:* Each customer order for any of the three products is represented by a product element. All three product elements can thus coexist in the process as in scenario III.

The process elements identified above are defined with their data models and control functions as shown in Fig. 4.3. We however omit these details and limit our focus onto their organisation and interaction behaviour.

7.4.2 Organisation of Process Elements

Similar to identification, the organisation of process elements mirrors their physical involvement in the process. In particular, each unit element is defined by the header elements that it is connected with, and each header elements by the unit or service elements that it connects together. In order to understand how this physical structure and interconnection of elements supports reconfigurability, we consider below how changing the flexibility available in the local design of unit, header or product elements can affect this property.

- *Unit Elements:* The capability of unit elements is varied between each being able to perform: (i) a single task, or (ii) multiple tasks, where a *task*, as defined in Section 5.2, refers to a unit operation (*e.g.*, reaction) with its associated materials and services.

Table 7.3. Variations in the organisation of process elements

	Recipe mapping Approach	Unit Element Capability	Physical Layout
Case 1	Product-centric	Single task	Fixed
Case 2	Unit-centric	Single task	Full
Case 3	Unit-centric	Multiple tasks	Flexible

- *Header Elements:* The capability of header elements is varied by changing their flexibility to interconnect process units between: (i) *fixed* connectivity, where connectivity is limited as in Fig. 7.1, (ii) *full* connectivity, where all unit elements can be connected to all other unit elements, and (iii) *flexible* connectivity, on the spectrum between fixed and full connectivity, where connectivity is enhanced by increasing the number of possible connections between unit elements in the fixed layout.
- *Product Elements:* The capability of product elements is varied by changing their their involvement in the process based on the approach used for recipe mapping, (see Section 5.2), *i.e.,* : (i) *product-centric approach*, where product elements are supplied with (non-linear) product recipes shown in Fig. 7.2, and (ii) *unit-centric approach*, where product elements are not supplied with any recipe at all, but this information is defined as part of the design of unit elements themselves. In the former, the product elements centrally assign processing tasks to unit elements, while in the latter the unit elements themselves select the tasks based on recipe information supplied to them.

The above variations thus entail different options by which the elements can be organised within the overall system. We consider, in particular, three such combinations characterised in Table 7.3 and represented in Fig. 7.4 (case 1), Fig. 7.5 (case 2) and Fig. 7.6 (case 3). The oblong symbols therein represent the unit elements, the rectangles represent the materials, and the lines connecting oblong and rectangle symbols are possible connections between unit elements. From these figures, we can make the following observations.

- In all three cases, it is assumed that the unit elements are not defined with any *a priori* information about other unit elements they may be connected with. They acquire this information from the associated header elements during synthesis phase.
- As compared to fixed layout in Fig. 7.4, the unit elements in the full (Figs. 7.5) or flexible (Fig. 7.6) layouts are defined with the exact set of materials they need to consume or produce. Embedding this additional information however does not fix the process schemes for either case as there exists multiple combinations of unit elements (in Fig. 7.5) or their

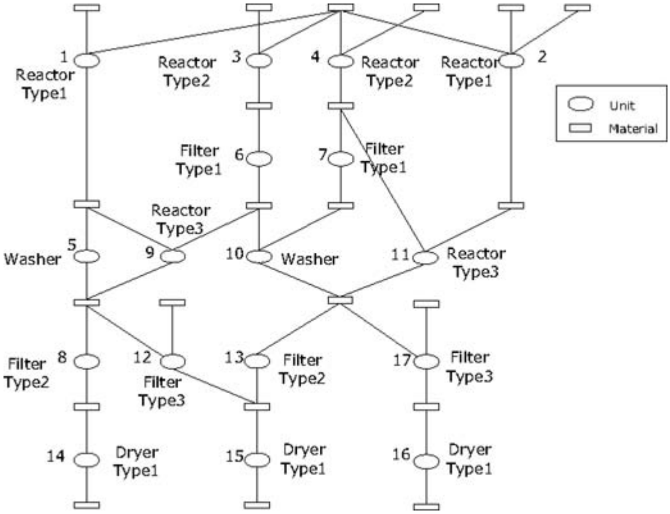


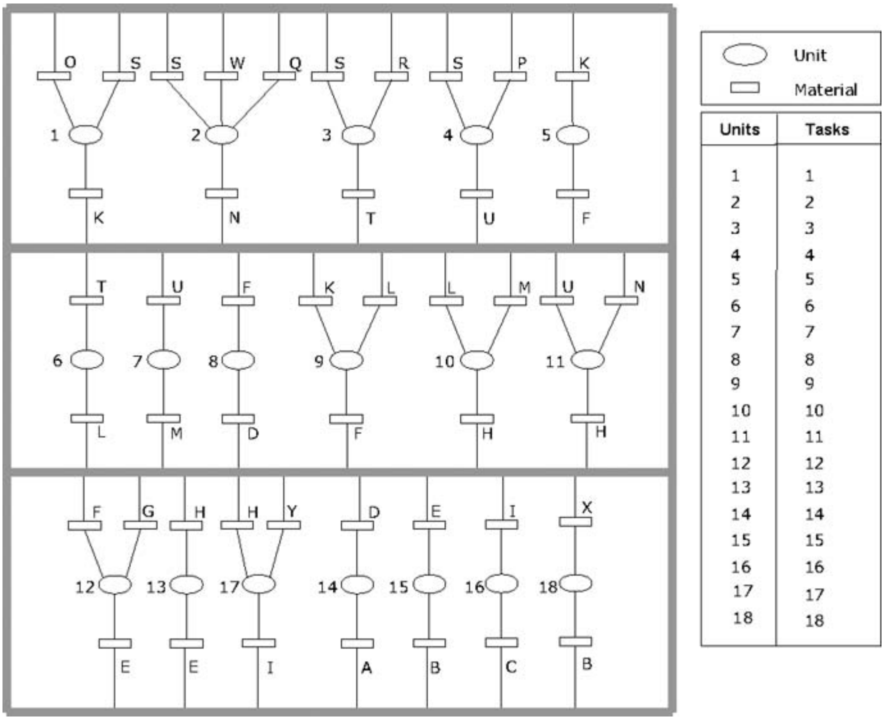
Fig. 7.4. Process layout: Case 1 - single task, fixed connectivity

selection of local tasks (in Fig. 7.6) which can produce the same end-product. The selection of a specific process scheme from these combinations occurs via distributed interactions between unit elements together with the associated product, header and service elements.

- In the full connectivity layout (Fig. 7.5), there exists no clear distinction between different header elements, rather the whole network can be seen as a single header element comprising multiple sub-networks connecting individual process stages. Note that a layout of this nature would be rare to find in reality however it shows the possibility of interconnections in so-called *pipeless* plants where the header elements can be thought as the material carrying equipment being moved around the plant.
- The flexible layout in Fig. 7.6 assumes that the unit elements are capable of performing multiple tasks. This feature in turn leads to a reduction in the unit element types from 18 in Figs. 7.4 or 7.5 to 8 in Fig. 7.6. As discussed later, this multipurpose capability combined with the flexible process layout in Fig. 7.6 helps enhance the reconfigurability of the control system to deal with changes or disturbances in a distributed way.

7.4.3 Interaction Behaviour of Process Elements

The process elements interact based on the interaction model presented in Chapter 5. With reference to the six scenarios described earlier, the *identify* phase leads to interpreting the effects of change or disturbance into specific requirements for reconfiguration. Where required (as in scenarios I, II, III), a new product element is also created to impose these requirements onto other



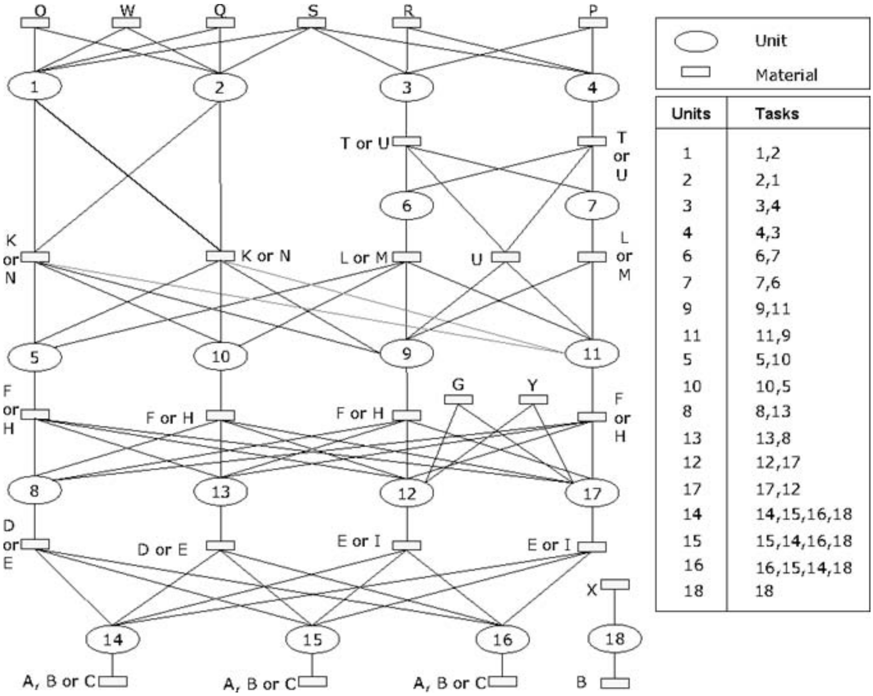
Units	Tasks
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18

Fig. 7.5. Process layout: Case 2 - single task, full connectivity

process elements. The *define* and *reconfigure* phases then involve the unit elements in the process together with associated header and service elements to build or amend the appropriate process scheme from possible interconnections. The production of the order starts during *operate* phase. On completion of the order the process scheme is terminated in the *terminate* phase (*e.g.*, in scenario III). During the *operate* phase, the process elements monitor plant conditions and invoke a new round of reconfiguration if a major failure or a disturbance is detected (*e.g.*, as in scenario IV where a unit element fails or in scenario VII where the demands for end-products change).

In what follows, we use the first five scenarios to describe the nature of interactions between process elements in all three cases in Table 7.3 individually. We assume that the above interaction sequence operates in background and focus only on the key interactions between unit and product elements and also the outcomes of these interactions in terms of the structure of resulting process schemes.

In the description we use the following notation: U followed by the number in process layout to refer to a unit element, PR followed by the product name to refer to a product element, and T followed by the number in Table 7.1 to refer to a task.



Units	Tasks
1	1,2
2	2,1
3	3,4
4	4,3
6	6,7
7	7,6
9	9,11
11	11,9
5	5,10
10	10,5
8	8,13
13	13,8
12	12,17
17	17,12
14	14,15,16,18
15	15,14,16,18
16	16,15,14,18
18	18

Fig. 7.6. Process layout: Case 3 - multiple tasks, flexible connectivity

Case 1: Single Task, Fixed Connectivity, Product-centric Approach

The first example demonstrates the interactions for Case 1 using the product-centric approach for recipe mapping in the define phase. We describe the operations required for scenarios I-V. As stated earlier, in a product-centric approach the product elements are supplied with product recipes shown in Fig. 7.2 and their role purely is to assign those tasks on the dark-lined sequence to suitable unit elements, but if this is not achievable an alternative sequence may be chosen.

I. *Start product B order:* On arrival of a new order for product *B*, a new product element PRB is created during identify phase. In the next define phase, PRB then engages with all unit elements in the process to perform recipe mapping. It announces each of its processing task on the preferred task sequence (*i.e.*, dark-lined sequence for product *B* in Fig. 7.2) and assigns them to suitable unit elements based on the responses received. If no response is received for any of the tasks, then it may choose a nearest alternative sequence in the recipe to minimise de-assignment of previously assigned tasks. The unit elements involved in the resulting tentative schemes then refine these schemes into a single scheme that is used for reconfiguration and operation. Fig. 7.7(a) shows the layout of the final scheme if this whole sequence is completed satisfactorily. Note that PRB only selects unit elements which match with its preferred sequence.

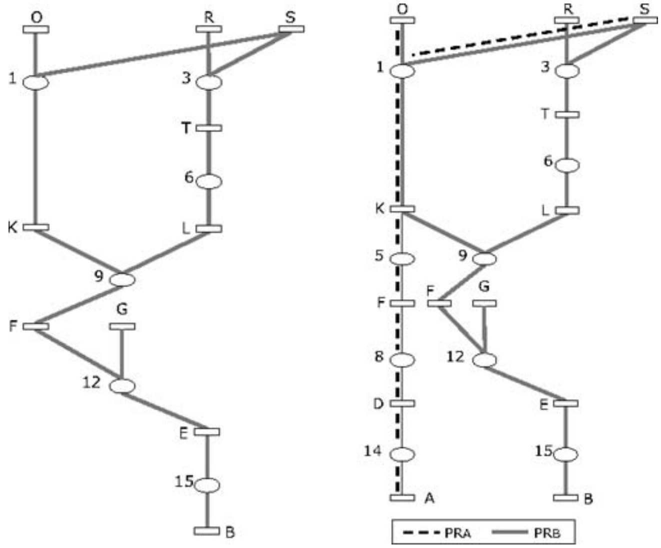
II. *Add product A order:* Next, a new product element PRA is created for product *A* which leads to a similar round of interactions involving all unit elements. Fig. 7.7(b) shows the process scheme resulting from these interactions. It now comprises both PRA and PRB. Note that although unit element U5 also produces material *F* for product *A* which is also an ingredient in producing *B*, it is in fact not involved in the production of *B* because the task T5 is not allowed on the preferred sequence of PRB. Unit element U1 is however involved in both schemes.

III. *Changeover from product B to C:* The changeover leads to creation of PRC and removal of PRB. Initially when PRC is created all unit elements in the process get involved in the creation of new process scheme for *C*, but as PRB is removed, those involved in producing *B* also terminate their tasks for PRB. Figs. 7.7(c) and 7.7(d) depict the resulting process schemes when all three product elements exist together and when only PRA and PRC remain.

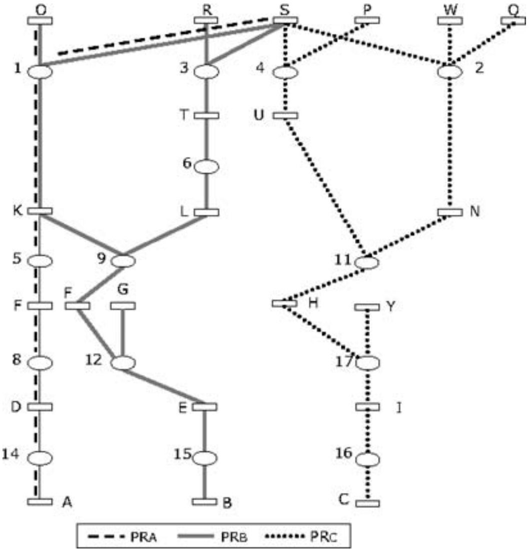
IV. *Unit element U11 fails:* Assume unit element U11 of type 'reactor type 1' fails and cannot supply material *H* any more. Thus, unit elements U17 and hence U16 also cannot continue with their tasks. An alternative source of *H* is thus required. Since no other unit elements on the preferred task sequence of PRC can supply *H*, PRC invokes a new round of interactions. During identify phase, the requirement imposed for PRC is to choose an alternative sequence that can produce *C*. The subsequent interactions then follow as in previous scenarios. Fig. 7.7(e) shows the process scheme based on a different sequence involving U10.

V. *Unit element U11 Rejoins:* The incoming unit element in this case announces its capability to all product and unit elements. Since PRC can make use of its facility to revert back to its preferred sequence, it has a choice whether to continue with the ongoing scheme or to choose this preferred option. Assuming the decision rule defined in PRC is to choose the preferred sequence where possible, it invokes a new round of interactions and reassigns the tasks as appropriate to return to the scheme in Fig. 7.7(d).

Note that unit elements in the above description only possess localised knowledge of their task capability (*i.e.*, the type of unit operation they can perform). They do not have the knowledge of preferred task sequence or the materials or services associated with the individual tasks. Such information is

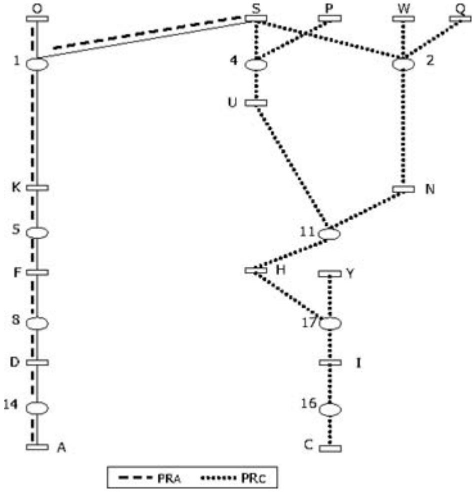


(a) I: Start product B (b) II: Add product A

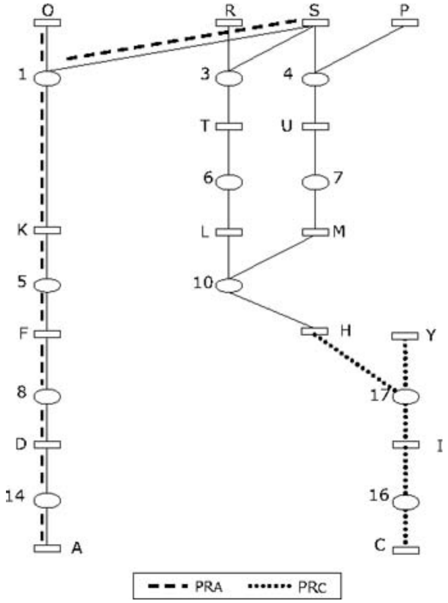


(c) III: Products A, B, C together

Fig. 7.7. Illustrations of process schemes: Case 1, Scenarios I, II, III



(d) III: Changeover from B to C



(e) IV: U11 fails

Fig. 7.7. Illustrations of process schemes: Case 1, Scenarios III, IV

assigned to them by the product elements. As a result unit elements are unable to respond to process disturbances (such as failure of U11) without interacting with product elements. This limitation is removed in the unit-centric approach as discussed in the next two cases.

Case 2: Single Task, Full Connectivity, Unit-centric Approach

Case 2 refers to full connectivity among unit elements which are now also defined with the exact set of materials for their tasks. Their role now involves finding from a large number of connections (due to full connectivity) a single process scheme that fits with the requirement of the product order. Below we describe the same scenarios (I-V) to describe how the interactions would proceed in this case.

I. *Start product B order:* On arrival of a new order for B, a product element PRB is created during identify phase. PRB is however not supplied with any recipe. Instead, the unit elements themselves identify the tasks they can to use to produce requested material. The interactions thus proceed in a demand-pull fashion starting from the end-product B. Since two unit elements, U15 and U18, can produce B, both initiate building a new process scheme. The build-up proceeds in the backward direction. Both units attempt to acquire the feedstock for their tasks (material E for T15 of U15 and material X for T18 of U18) from upstream unit elements. All unit elements which can supply these feedstocks get involved. U18 will however find that no other unit element in the process can supply X. It therefore cannot involve in producing B. For U15, the interactions proceed further. Fig. 7.8(a) shows the process scheme that results from these interactions after the synthesis phase is completed.

   It can be noted that, unlike Fig. 7.7(a) in Case 1, the final scheme includes all unit elements which could involve in producing B as there are no constraints on the task selection from product recipe. Note also that materials L and U are used by multiple unit elements – material L used by U9 and U10 and material U by U7 and U11. These materials thus fall along two different branches of the same process scheme that lead to product B. Thus, if a unit element in either branch fails, the unit elements in the other branch should be able to take over its load within their capacity (see scenario IV).

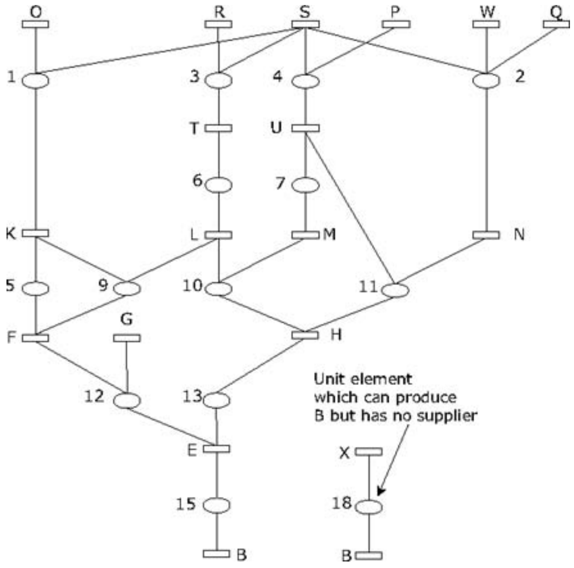
II. *Add product A order:* Arrival of a new product order for A leads to creation of product element PRA and a new round of interactions. Since only U14 can produce A, it initiates the formation of a process scheme. The interactions proceed similar to previous scenario. Fig. 7.8(b) shows the final process scheme. Note that material F is now involved in both process schemes. Thus, when U8 makes its request for F, both U5 and U9, which are already engaged in producing B, also engage in producing A. These elements subsequently reallocate their feedstock demands for K and L by re-interacting with upstream supplier unit elements. The effects of this reallocation incrementally propagates to other unit elements in both process schemes.

III. *Changeover from product B to C:* Changeover from product B to C leads to creation of PRc and removal of PRb. Fig. 7.8(c) and 7.8(d) show the process schemes when all three product elements coexist and when only PRa and PRc remain. In total, two new unit elements U16 and U17 get involved while U12, U13 and U15 are removed. Subsequently, the latter three elements also terminate their interactions for material E and then for materials F and H. The unit elements upstream reallocate their material demands accordingly.

IV. *Unit Element U11 Fails:* Failure of U11 leads to an abrupt termination of all its interactions with upstream and downstream unit elements. U2 will thus also be removed from process scheme for product C. Since U10 also supplies material H, it takes over the load from U11 within its capacity and reallocates its feedstock demands as appropriate. This response to failure emerges directly by the interactions between failed element U11, and the affected elements U17 and U10. The resulting scheme from these interactions is not shown here for brevity, but its structure can be easily derived from Fig. 7.8(d).

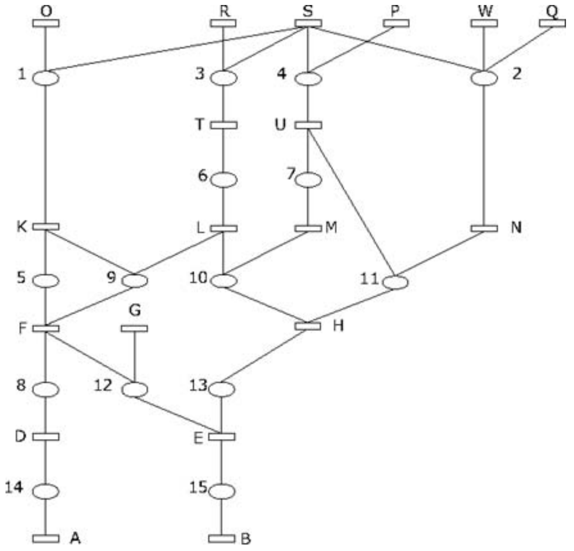
V. *Add a New Unit U19 or U11 Rejoins:* In either case the incoming unit element announces its presence to other unit elements in terms of the materials it can supply. Unit elements which can use this facility, *e.g.*, U17 here, then interact with it to reallocate its feedstock demands accordingly. The interactions should thus reinstate the scheme in Fig. 7.8(d).

It can be seen that by supplying material-specific information for their tasks, the unit elements are able to manage recipe mapping activity in a distributed manner. This distribution helps manage a change or failure in a graceful manner compared to product-centric approach in Case 1. More importantly, the unit elements are also capable of selecting processing tasks that are known locally that otherwise may not be specified by the developers of product recipes situated often remotely. A benefit of this can be seen by comparing Fig. 7.7(a) with Fig. 7.8(a). In the former only those unit elements whose tasks match with the recipe are selected, while in the latter all unit elements which can involve in making B are selected. The latter is thus also likely to have a better chance to respond to a change or failure than the former.



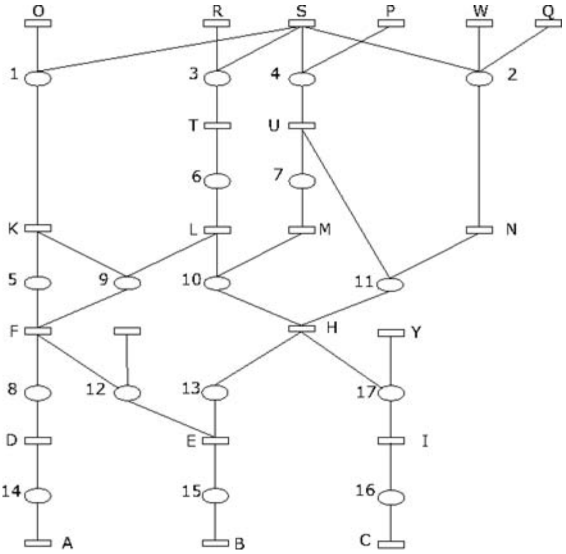
Unit element which can produce B but has no supplier

(a) I: Start product *B* order

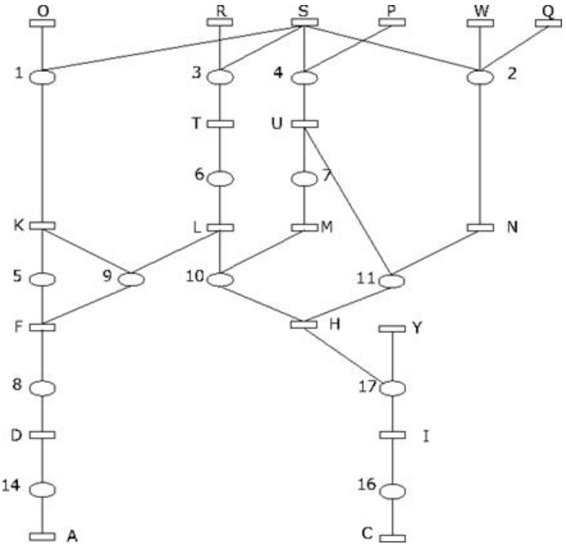


(b) II: Add product *A* order

Fig. 7.8. Illustration of process schemes: Case 2, Scenarios I, II



(c) III: Products *A*, *B* and *C* together



(d) III: Changeover from product *B* to *C*

Fig. 7.8. Illustration of process schemes: Case 2, Scenarios III, IV

Case 3: Multiple Tasks, Flexible Connectivity, Unit-Centric Approach

Case 3 (Fig. 7.6) removes the limitation of single task capability and also considers flexible connectivity among unit elements. The unit elements now receive a combined choice of selecting local tasks and/or the supplier elements for their feedstocks in developing the process schemes. As the description of scenarios I-V next illustrate, this flexibility leads to an added freedom in responding to emerging changes or disturbances than the previous two cases.

I. *Start product B order:* On arrival of product order for B, a new order PRB is similarly created with a new round of demand-pull interactions. Four unit elements, U14, U15, U16 and U18, can produce B. For U18 the scheme cannot proceed as no unit element can supply X. For the other three elements the interactions proceed further. U8, U13, U12 and U17 can supply E: U8 and U13 via task T13 and U12 and U17 via T12. All four hence get involved and try to extend the process scheme by acquiring their feedstock H for T13, and $\{F, G\}$ for T12. Note that there are multiple task sequences available for producing both F and H. Unit elements U5 and U10 can use task T5 to produce F and T10 for H. Similarly U9 and U11 can use task T9 for F and T11 for H. These unit elements thus face a choice when they attempt to acquire feedstocks for these alternative tasks. The final selection of a specific task would occur during synthesis phase when unit elements refine these tentative process schemes into a single scheme. To simplify the discussion, we consider an assumption that the unit elements select the first task in the sequence in Fig. 7.6 when they have such a choice, *i.e.,* U5 and U10 select tasks T5 and T10 respectively, while U9 and U11 select tasks T9 and T11. The complete process scheme thus developed involves six more unit elements from upstream. We do not show the resulting scheme for brevity. We can see however that compared to Cases 1 or 2, the introduction of multipurpose character of unit elements with flexible connections leads to increased choice available to unit elements for producing B.

II. *Add product A:* On arrival of product A order a new product element PRA is created which announces its requirements. Unit elements U14, U15 and U16 which all can produce A are however engaged with PRB. These unit elements, while continuing with their tasks, initiate a new round of interactions to develop the tentative schemes for A. During synthesis phase these elements then use a production goal to decide whether or not to de-commit from their existing tasks and involve in the production of A. For simplicity of discussion, we assume that U14 prefers the first task (*i.e.,* T14) in sequence in Fig. 7.6 over T15. It hence de-commits from T15. Its capacity for producing B is transferred to U15 and U16 as appropriate. The same decision rule also extends to unit element U8 which de-commits from its task to involve in the process scheme for A.

III. *Changeover from product B to C:* Creation of PRc leads to U16 changing its task from T15 to T16 (by continuing with the assumption of task preference). Unit element U15 is now the only element producing B. The same change also occurs for U17 which changes from T12 to T17. Subsequently, when PRb is removed, all unit elements which were engaged with product B de-commit from their tasks. These unit elements now become available and announce their capabilities to other unit elements. The process schemes for products A and C are thus revised to involve these unit elements as appropriate.

IV. *Unit Element U11 fails:* The failure of U11 results in a partial loss of supply for material H. U10 which is also involved in producing H attempts to take over its load through T10. It is possible here that U9 could replace U11 if material H is more essential than F when comparing the importance of product C to product A or if the capacity for F can be shifted to other unit elements in the process scheme for A. The unit elements make these decisions during synthesis phase in deciding the final process scheme and their local operating settings.

V. *Add a New Unit U19 or U11 Rejoins:* The new element announces its capabilities and gets involved in the interactions. If U9 has changed its task in the previous scenario than it has a choice to revert back to its original task since an alternative supplier of H is available. Using the rule of task preference, it will do so. The outcome of the interactions should lead to reinstating the scheme in Scenario III.

As we can see, the enhancement in local capabilities of unit elements aided by the flexibility in their interconnections leads to an increased choice and reconfigurability in all scenarios described above compared to Cases 1 and 2. This observation thus provides a crucial guideline in organising the process elements based on the distributed architecture and the interaction model discussed in previous chapters.

7.4.4 Coordination of Distributed Operating Settings

The coordination of local and network parameters of process elements occurs via their distributed interactions. During synthesis of a process scheme from multiple tentative schemes this coordination involves various mixed-integer decisions such as which tasks and hence supplier elements should be selected (as discussed in Case 3). While a complete computational framework covering all such decisions is beyond the scope of this text, the algorithm presented in Chapter 6 provides a sensible framework to define these interactions in a mathematical form.

Below, we illustrate the developments in Chapter 6 by applying them to the current example, in particular to scenario VI. We use the layout in

Fig. 7.8(c) which includes all 17 unit elements involved in making A, B, and C. It is assumed that the unit elements have found this process scheme during synthesis phase and the aim of distributed algorithm is to find the settings of local and interaction parameters.

Modelling the Local Dynamics of Process Elements

The local dynamics and interconnections of unit elements are modelled using the linear, dynamical model presented in Section 6.2. In particular, the demands for outgoing products of a unit element are treated as disturbances and the demands that it places for materials and services to upstream unit elements and service elements are treated as the manipulated variables that it controls. Tables C.9 and C.10 in Appendix C define the problem data for all 17 units in Fig. 7.8(c). The problem data is implemented using the framework defined in Appendix B.

Operation of the Distributed Algorithm

Each of the 17 unit elements is supplied with the generic unit software module defined in Appendix B. As stated therein, the module is generic in that it applies to all four types of junction block connections of a unit element. Depending on the type of junction block generated in the synthesis phase, the optimality cuts generated are varied as appropriate.

To model scenario VI, we assume that all three products initially have a demand of 10 deviation units from a nominal set-point. Tables C.11 and C.12 in Appendix C summarise the progress of the distributed algorithm for local state and manipulated variables $x_{i,z}$ and $u_{i,z}$ for unit element $i = 1, \ldots, 17$, where z refers to z^{th} element of x_i and u_i for unit element i. Note that we have numbered unit elements by $i = 1, \ldots, 17$ instead of the ordering (n, s) in Chapter 6. It can be seen that the algorithm converges to optimal solution within three or four iterations, although an accuracy of four digits requires further iterations.

As a next step to scenario VI, the demand for all three products is changed from 10 to 20 deviation units after iteration 10. Fig. 7.9 shows the effects of approximate cut updates on the sub-problems of unit elements 14 and 16. Fig. 7.10 demonstrates the effects of this change in demands in terms of the feedstock demands that unit elements 14 and 16 place to their upstream units. The solution algorithm is able to absorb this change and converge to a new optimum solution after 21 iterations.

Fig. 7.11(a) on Page 148 shows the computational performance of the distributed algorithm in terms of floating-point calculations (flops) required as compared to a centralised algorithm for solving a series of 30 different data-sets for the same problem. In the case of the distributed algorithm, we terminate the algorithm if the number of iterations reaches more than 20. We can observe that the distributed algorithm, although not as efficient (which is

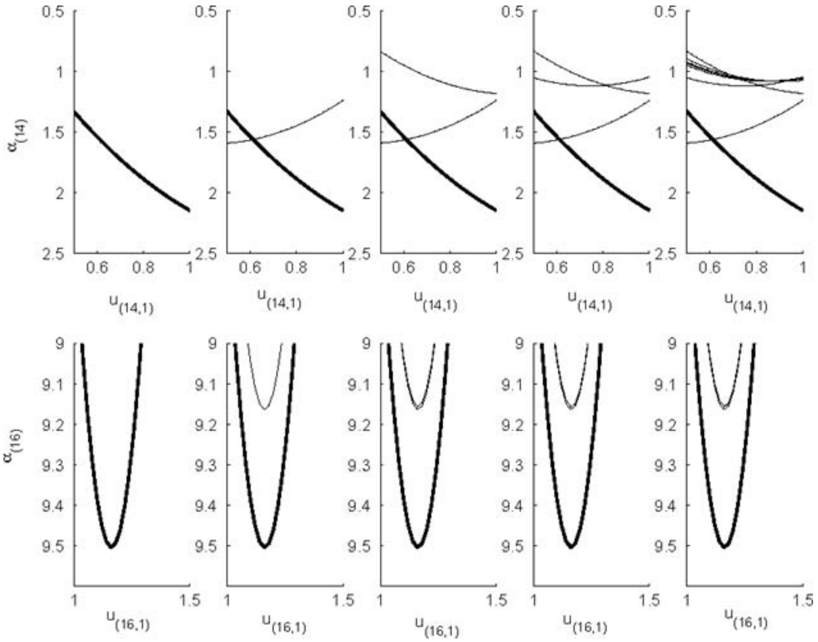


Fig. 7.9. Effects of approximate cut updates on the value functions of unit elements 14 and 16 sub-problems

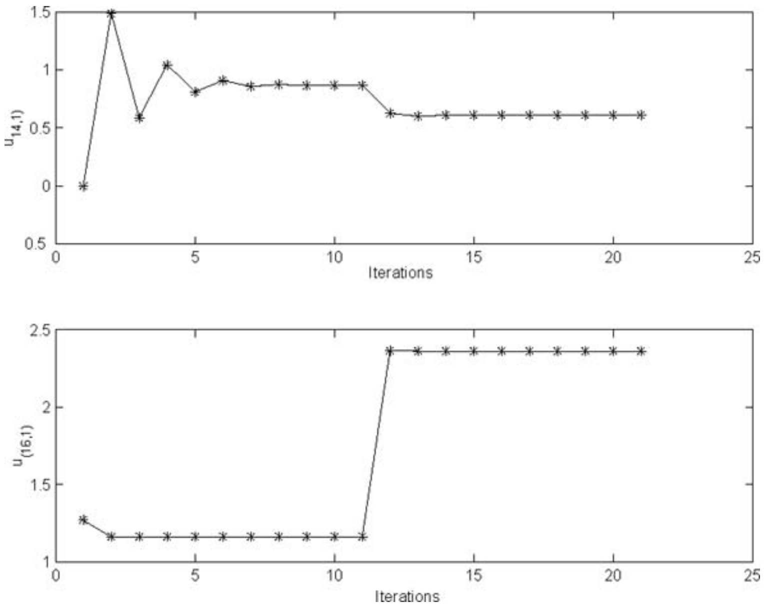


Fig. 7.10. Effects of change in terminal demands of unit elements 14 and 16

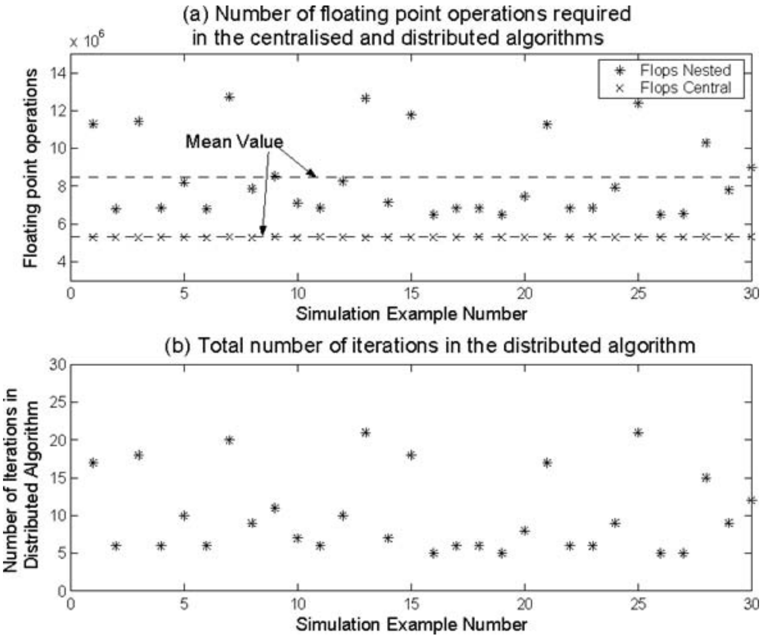


Fig. 7.11. (a) Comparison of floating -point operations (flops) required between centralised and distributed algorithms; (b) Number of iterations required in distributed algorithm, both for 30 data-sets

not the aim for reconfigurable control), compares well to centralised algorithm in most cases. Fig. 7.11(b) shows the number of iterations required for solving all 30 problems in the distributed algorithm. Again, the iterations remain limited and within a range of 5 to 20.

7.5 Summary

This chapter considered a case example of a multipurpose process plant to illustrate the reconfigurable process control developments from the previous three chapters. The discussions in the chapter have clearly highlighted the nature of bottom-up response of process elements under changing conditions which should be compared to those of a conventional system where the same response would be derived by a higher-level scheduler or optimiser. We emphasise that in DRPC this response is not pre-defined in any of the three cases considered, but rather it emerges from the localised decision rules of the product and unit elements involved and a global method for coordinating these decisions through the interaction model.