Centralized–decentralized optimization for refinery scheduling

Nikisha Shah, Georgios K.D. Saharidis, Zhenya Jia, Marianthi G. Ierapetritou*

Department of Chemical and Biochemical Engineering, Rutgers University, Piscataway, NJ 08854, USA

A R T I C L E   I N F O

Article history:
Received 6 November 2008
Received in revised form 14 May 2009
Accepted 3 June 2009
Available online 11 June 2009

Keywords:
Refinery
Scheduling
Optimization
Centralized
Decentralized

A B S T R A C T

This paper presents a novel decomposition strategy for solving large scale refinery scheduling problems. Instead of formulating one huge and unsolvable MILP or MINLP for centralized problem, we propose a general decomposition scheme that generates smaller sub-systems that can be solved to global optimality. The original problem is decomposed at intermediate storage tanks such that inlet and outlet streams of the tank belong to the different sub-systems. Following the decomposition, each decentralized problem is solved to optimality and the solution to the original problem is obtained by integrating the optimal schedule of each sub-systems. Different case studies of refinery scheduling are presented to illustrate the applicability and effectiveness of the proposed decentralized strategy. The conditions under which these two types of optimization strategies (centralized and decentralized) give the same optimal result are discussed.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Production scheduling defines which products should be produced and which products should be consumed in each time instant over a given small time horizon; hence, it defines which run-mode to use and when to perform changeovers in order to meet the market needs and satisfy the demand. Large-scale scheduling problems arise frequently in oil refineries where the main objective is to assign sequence of tasks to processing units within certain time frame such that demand of each product is satisfied before its due date. As the scale of the production problem increases, the mathematical complexity of the corresponding scheduling problem increases exponentially. Decomposition of the initial system into sub-systems which are easier to be solved, is a natural way to deal with this type of optimization problems.

There are relatively few papers that have addressed planning and scheduling problems using centralized and decentralized optimization strategies providing a comparison of these two approaches. Kelly and Zyngier (2008) presented a procedure to find a suitable way to decompose large decision-making problems and compared different decentralized approaches using hierarchical decomposition heuristics. The focus of their work was to find globally feasible solutions to large decentralized and distributed decision-making problems when a centralized approach is not possible. Saharidis, Dallery, and Karaesmen (2006) and Saharidis, Kouikoglou, and Dallery (2009) studied the problem of production planning in deterministic and stochastic environments and compared centralized and decentralized optimization for an enterprise consisting of two production plants in series producing many different outputs with subcontracting options. Chen and Chen (2005) studied a joint replenishment arrangement with a two-echelon supply chain with one supplier and one buyer, facing a deterministic demand and selling a number of products in the marketplace. They proposed both centralized and decentralized decision policies to analyze the interplay and to investigate the joint effects of two-echelon coordination and multi-product replenishment on the reduction of total costs. The cost differences between these policies show that the centralized policy significantly outperforms the decentralized policy. Cioni, Lavagnilio, Bassetti, Mummolo, and Leva (2003) present a case study from the automotive industry dealing with the lot sizing and scheduling decisions in a multi-site manufacturing system. They use a hybrid approach which combines mixed-integer linear programming model and simulation to test local and global production strategies. Their results show that local optimization strategy allows a cost reduction of about 19% compared to the reference actual annual production plan, whereas the global strategy leads to a further cost reduction of 3.5% and a better overall economic performance. Harjunkoski and Grossmann (2001) presented a decomposition scheme for solving large scheduling problems for steel production which splits the original problem into sub-systems using the special features of steel making. Their proposed approach cannot guarantee global optimality, but comparison with theoretical estimates indicates that the method produces solutions within 1–3% of the global optimum. Bassett, Pekny, and Reklaitis (1996) presented resource decomposition method to reduce problem complexity by dividing the scheduling problem into sub-sections based on its process recipes. They showed that the overall solution time using resource
In this work, the problem of refinery scheduling optimization is addressed with centralized and decentralized decision making processes. The paper is organized as follows. Section 2 describes general structure of problem studied in this paper. Section 3 defines the mathematical formulation of the problem, whereas the decomposition approach is presented in Section 4. Section 5 presents a real case study provided by Honeywell Hi-Spec Solutions and provides comparative results for centralized and decentralized optimization of the system. Finally Section 6 draws conclusions indicating perspectives for future research.

2. Problem definition

In general there are two decision levels in refinery process operations—the planning and the scheduling level. The planning level determines the volume of raw materials needed for the upcoming months (typically 12 months), and the type of final products and the estimated quantities to be ordered, depending on demand forecasts. After determining the yearly plan in the second level we have to determine the optimal production scheduling. The scheduling level determines the detailed schedule of each CDU and other production unit for a shorter period (typically 10 days) by taking into account the operational constraints of the system under study. Once the plan is known (the quantities and the types of final products ordered as well as the arrival of raw materials), managers must schedule the production of each unit based on the

Nomenclature

<table>
<thead>
<tr>
<th>Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>j</td>
</tr>
<tr>
<td>jst</td>
</tr>
<tr>
<td>i</td>
</tr>
<tr>
<td>n</td>
</tr>
<tr>
<td>s</td>
</tr>
<tr>
<td>k</td>
</tr>
<tr>
<td>K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>j</td>
</tr>
<tr>
<td>jst</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>i(i)</td>
</tr>
<tr>
<td>jst(i)</td>
</tr>
<tr>
<td>stseq(i)</td>
</tr>
<tr>
<td>jstprod(jst)</td>
</tr>
<tr>
<td>fprod(jst)</td>
</tr>
<tr>
<td>jstseq(s)</td>
</tr>
<tr>
<td>jstprod(s)</td>
</tr>
<tr>
<td>jstin(s,k)</td>
</tr>
<tr>
<td>jstseq(s)</td>
</tr>
<tr>
<td>jstprod(s)</td>
</tr>
<tr>
<td>jstseq(s,k)</td>
</tr>
<tr>
<td>jstprod(s,k)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rmin(i,j)</td>
</tr>
<tr>
<td>Rmax(i,j)</td>
</tr>
<tr>
<td>Pmin(s)</td>
</tr>
<tr>
<td>Pmax(s)</td>
</tr>
<tr>
<td>Cmax(s)</td>
</tr>
<tr>
<td>Vmax(jst)</td>
</tr>
<tr>
<td>p^i(s)</td>
</tr>
<tr>
<td>p^j(s)</td>
</tr>
<tr>
<td>d(s)</td>
</tr>
<tr>
<td>r(s,n)</td>
</tr>
<tr>
<td>r'(s,n)</td>
</tr>
<tr>
<td>stin(jst)</td>
</tr>
<tr>
<td>UH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>w(i,j,n)</td>
</tr>
<tr>
<td>iter</td>
</tr>
<tr>
<td>b(i,j,n)</td>
</tr>
<tr>
<td>st(jst,n)</td>
</tr>
<tr>
<td>inflow1(jst,n)</td>
</tr>
<tr>
<td>outflow1(jst,n)</td>
</tr>
<tr>
<td>inflow2(s,j,n)</td>
</tr>
<tr>
<td>outflow2(s,j,n)</td>
</tr>
<tr>
<td>inflow(jst,n)</td>
</tr>
<tr>
<td>outflow(jst,j,n)</td>
</tr>
<tr>
<td>in(jst,n)</td>
</tr>
<tr>
<td>out(jst,j,n)</td>
</tr>
<tr>
<td>unitflow(s,j,j,n)</td>
</tr>
<tr>
<td>st(jst,n)</td>
</tr>
<tr>
<td>Ts(j,n)</td>
</tr>
<tr>
<td>Tf(j,n)</td>
</tr>
<tr>
<td>Tss(i,jst,n)</td>
</tr>
<tr>
<td>Tsf(i,jst,n)</td>
</tr>
<tr>
<td>Tss(jst,n)</td>
</tr>
<tr>
<td>Tsf(jst,n)</td>
</tr>
</tbody>
</table>

Decomposition is significantly lower than the time needed to solve the global problem. However, their proposed resource decomposition method did not involve any feedback mechanism to incorporate “raw material” availability between sub-problems.
3. Mathematical formulation

In this section a mathematical model is presented for the scheduling of the refinery system presented in Section 2 where the objective is to minimize the overall makespan. The developed mathematical formulation uses continuous time representation because this leads to reduced number of decision variables and constraints compared to the discrete time representation; and due to the continuous operating mode, continuous time representation can provide more accurate results (Ierapetritou, Hene, & Floudas, 1999; Jia, Ierapetritou, & Kelly, 2003). The mathematical formulation proposed in this paper has the following main assumptions: (1) the change over time between different tasks at each unit is negligible; and (2) the processes are running at steady state. The proposed model presented in this section minimizes the overall makespan of the refinery production and involves allocation, capacity, storage, material balance, demand, and time sequence constraints. The integer and continuous decision variables used in the developed model give rise to a mixed integer linear programming (MILP) problem (see Nomenclature).

3.1. Mathematical model constraints

For the minimization of the makespan, four main operating rules have to be followed. The first one is to satisfy the constraint that at most one task can take place in one production unit at one time interval and that at most one material can be stored in one storage tank at one time interval (constraints (1)). The second one is to satisfy the material balance constraints (constraints (7)–(10)). The third one guarantees the demand satisfaction (constraint (11)). Finally the forth one guarantees the correct time sequence of the tasks and continuous operation of production units given that a continuous time representation is used (constraints (12)–(35)).

3.1.1. Allocation constraints

Constraints (1) express that if a task \( i \) starts at event point \( n \), then it must be performed in one of the suitable units \( j \). It also satisfies the operating rule that a unit can physically perform only one task at any given time.

\[
\sum_{i \in I(j)} w_t(i, j, n) \leq 1, \quad \forall j \in J, n \in N
\]  

3.1.2. Capacity constraints

Constraints (2) enforce the requirement that material processed by unit \( j \) performing task \( i \) at any point \( n \) is bounded by the maximum and minimum rates of production. The maximum and minimum production rates multiply by the duration of task \( i \) performed at unit \( j \) give the maximum and minimum material being

\[
\sum_{i \in I(j)} v_t(i, j, n) \leq w_t(i, j, n) u_t(i, j, n), \quad \forall j \in J, n \in N
\]
processed by unit \(j\) correspondingly.

\[
\begin{align*}
&\rho^{\min}(i, j) \times (TF(i, j, n) - Ts(i, j, n)) \leq b(i, j, n) \leq \rho^{\max}(i, j, n) \\
&- Ts(i, j, n), \quad \forall i \in I, j \in J, n \in N
\end{align*}
\] (2)

3.1.3. Storage constraints

Variable \(in(j, jst, n)\) is equal to 1 if there is flow of material from production unit \((i)\) to storage tank \((jst)\) at event point \((n)\); otherwise it is equal to 0. Variable \(out(jst, j, n)\) is equal to 1 if material is flowing from storage \((jst)\) to unit \((j)\) at event point \((n)\), otherwise it is equal to 0. Eqs. (3) and (4) are capacity constraints for storage tank and they define the binary variables associated with flow in and out of storage tanks. Constraints (3) state that if there is material inflow to tank \((jst)\) at interval \((n)\) then total amount of material inflow to the tank should not exceed the maximum storage capacity limit. Similarly, constraints (4) are for outflow from tank.

\[
\begin{align*}
&\text{inflow}(j, jst, n) \leq V^{\max}(jst) \times \text{in}(j, jst, n), \\
&\forall j \in J, jst \in JSTprod(j), n \in N
\end{align*}
\] (3)

\[
\begin{align*}
&\text{outflow}(jst, j, n) \leq V^{\max}(jst) \times \text{out}(jst, j, n), \\
&\forall j \in J, jst \in JSTprod(j), n \in N
\end{align*}
\] (4)

Constraints (5) and (6) represent the requirement that the material in tank \((jst)\) should not exceed the capacity limit \(V^{\max}(jst)\) of this storage tank at any event point \((n)\).

\[
\begin{align*}
&st(jst, n) + \sum_{j \in JSTprod(jst)} \text{inflow}(j, jst, n) + \text{inflow1}(jst, n) \\
&\leq V^{\max}(jst), \quad \forall jst \in Jst(s), n \in N
\end{align*}
\] (5)

\[
\begin{align*}
&\text{stin}(jst) + \sum_{j \in JSTprod(jst)} \text{inflow}(j, jst, n) + \text{inflow1}(jst, n) \\
&\leq V^{\max}(jst), \quad \forall jst \in Jst(s), n = 0
\end{align*}
\] (6)

3.1.4. Material balance constraints for operating unit

Constraints (7) represent the requirement that production of a unit should be equal to the sum of the amount of flows entering its subsequent storage tanks and reactors, and the delivery to the market.

\[
\begin{align*}
&\sum_{s \in S} \rho^p(s, i) \times b(i, j, n) = \sum_{jst \in JSTprod(jst) \cap Jst(s)} \text{inflow}(j, jst, n) + \text{unitflow}(s, j, j', n) + \text{outflow2}(s, j, n), \\
&\forall s \in S, j \in J, n \in N
\end{align*}
\] (7)

Similarly, constraints (8) represent that consumption at a unit is equal to sum of the amount of streams coming from preceding storage tanks, previous units, and stream coming from supply.

\[
\begin{align*}
&\sum_{s \in S} \rho^c(s, i) \times b(i, j, n) = \sum_{jst \in JSTprod(jst) \cap Jst(s)} \text{outflow}(jst, j, n) + \text{unitflow}(s, j, j', n) + \text{outflow2}(s, j, n), \\
&\forall j \in J, s \in S, n \in N
\end{align*}
\] (8)

3.1.5. Material balance constraints for storage tank

Similar to material balance constraints for units, the material balance constraints (9) and (10) for the storage tanks state that the inventory of a storage tank at one event point is equal to that at previous event point adjusted by the input and output stream amount.

\[
\begin{align*}
&\text{St}(jst, n) = \text{St}(jst, n - 1) + \sum_{j \in JSTprod(jst)} \text{inflow}(j, jst, n) + \text{inflow1}(jst, n) - \sum_{j \in JSTprod(jst)} \text{outflow}(jst, j, n) - \text{outflow1}(jst, n), \\
&\forall jst \in Jst(s), j \in J, n \in N
\end{align*}
\] (9)

\[
\begin{align*}
&\text{St}(jst, n) = \text{St}(jst) + \sum_{j \in JSTprod(jst)} \text{inflow}(j, jst, n) + \text{inflow1}(jst, n) - \sum_{j \in JSTprod(jst)} \text{outflow}(jst, j, n) - \text{outflow1}(jst, n), \\
&\forall jst \in Jst(s), n \in N
\end{align*}
\] (10)

3.1.6. Demand constraints

Demand for each finished final product \(d(s)\) must be satisfied in centralized problem and also in decentralized problem. Constraints (11) state that production units must at least produce enough material to satisfy the demand by the end of the time horizon.

\[
\begin{align*}
&\sum_{(jst, n) \in Jst(s)} \text{outflow1}(jst, n) + \sum_{(j, n) \in J} \text{outflow2}(s, j, n) \geq d(s), \\
&\forall j \in J, s \in S
\end{align*}
\] (11)

3.1.7. Duration constraints

3.1.7.1. Time sequence constraints for each unit

Constraints (12)–(14) express that if task \((i)\) starts at event point \((n + 1)\), then it must start after the end of the same task happening at event point \((n)\) in the same unit \((i)\). If task \((i)\) takes place at unit \((j)\) at event point \((n)\) then \(wv(i, j, n) = 1\), and \(Ts(i, j, n + 1)\) must be greater than or equal to \(Ts(i, j, n)\). If \(wv(i, j, n) = 0\) then the constraint in Eq. (12) is relaxed and constraints in Eq. (13) and (14) enforce the sequencing of tasks.

\[
\begin{align*}
&Ts(i, j, n + 1) \geq TF(i, j, n) - UH(1 - wv(i, j, n)), \\
&\forall i \in I, j \in J(i), n \in N
\end{align*}
\] (12)

\[
\begin{align*}
&Ts(i, j, n + 1) \geq Ts(i, j, n), \\
&\forall i \in I, j \in J(i), n \in N
\end{align*}
\] (13)

\[
\begin{align*}
&TF(i, j, n + 1) \geq TF(i, j, n), \\
&\forall i \in I, j \in J(i), n \in N
\end{align*}
\] (14)

Constraints (15) represent the rule that if task \((i')\) should happen at event point \((n)\) in unit \((j)\) then task \((i)\) must start at event point \((n + 1)\) after the end of task \((i')\) at event point \((n)\).

\[
\begin{align*}
&Ts(i, j, n + 1) \geq TF(i', j, n) - UH(1 - wv(i', j, n)), \\
&\forall j \in J(i), i \in I(j), i' \in I(j), i \neq i', n \in N
\end{align*}
\] (15)

Constraints (16)–(19) represent that two consecutive productions, where unit \((j)\) consumes the material produce by unit \((j')\), with no storage in between, happen at the same time since production units operate as continuous processes. If \(wv(i, j, n) = 1\) and \(wv(i', j, n) = 0\) then \(Ts(i, j, n) = Ts(i', j, n)\) and \(TF(i, j, n) = TF(i', j, n)\). If either \(wv(i, j, n) = 0\) or \(wv(i', j, n) = 0\), then the constraints are relaxed.
Ts(i, j, n) ≤ Ts(i′, j′, n) + UH(2 − w(i, j, n) − w(i′, j′, n)),
∀ j′ ∈ J, j ∈ Jseq(j′), i ∈ I(j), i′ ∈ I(j′), n ∈ N
(16)

Ts(i, j, n) ≥ Ts(i′, j′, n) − UH(2 − w(i, j, n) − w(i′, j′, n)),
∀ j′ ∈ J, j ∈ Jseq(j′), i ∈ I(j), i′ ∈ I(j′), n ∈ N
(17)

Tj(i, j, n) ≤ Tj(i′, j′, n) + UH(2 − w(i, j, n) − w(i′, j′, n)),
∀ j′ ∈ J, j ∈ Jseq(j′), i ∈ I(j), i′ ∈ I(j′), n ∈ N
(18)

Tj(i, j, n) ≥ Tj(i′, j′, n) − UH(2 − w(i, j, n) − w(i′, j′, n)),
∀ j′ ∈ J, j ∈ Jseq(j′), i ∈ I(j), i′ ∈ I(j′), n ∈ N
(19)

3.1.7.2. Time sequence constraints connecting unit and storage tank. Constraints (20)–(23) state that production and storage occur at the same time. If unit (j) produces the material that is stored in tank (jst), then start and finishing time of production task at unit (j) and inflow to the storage tank must be same.

Ts(i, j, n) ≤ Tst(j, jst, n) + UH(2 − w(i, j, n) − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(20)

Ts(i, j, n) ≥ Tst(j, jst, n) − UH(2 − w(i, j, n) − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(21)

Tj(i, j, n) ≤ Tst(j, jst, n) + UH(2 − w(i, j, n) − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(22)

Tj(i, j, n) ≥ Tst(j, jst, n) − UH(2 − w(i, j, n) − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(23)

Constraints are given by Eqs. (24)–(27) state that storage and production happen at the same time. If tank (jst) stores the material that is consumed in the unit (j), then outflow from storage tank and production take place at the same time.

Ts(i, j, n) ≤ Tst(j, jst, n) + UH(2 − w(i, j, n) − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(24)

Ts(i, j, n) ≥ Tst(j, jst, n) − UH(2 − w(i, j, n) − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(25)

Tj(i, j, n) ≤ Tst(j, jst, n) + UH(2 − w(i, j, n) − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(26)

Tj(i, j, n) ≥ Tst(j, jst, n) − UH(2 − w(i, j, n) − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), i ∈ I(j), n ∈ N
(27)

3.1.7.3. Sequence constraints for storage tank. Constraints are given in Eqs. (28)–(35) connect input and output flow happening at event (n) to the next event point (n + 1) for storage tanks. Input starts at event point (n + 1) after output finishes at event point (n) and output flow happens at event point (n + 1) after the end of input flow happening at event point (n). Binary variable in(jst,n) equal to 1 when material is flowing into the tank (jst) from unit (j) at event point (n), otherwise its zero. Similarly out(jst,n) equal to 1 when material is flowing from the tank to the unit at event point (n), otherwise its zero.

Constraints (28), (31), (34) and (35) are active when the binary variables are equal to 1, whereas when the binary variables are equal to 0, the sequence constraints are enforced by Eqs. (29), (30), (32) and (33).

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(28)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − jstprod(jst), n ∈ N
(29)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(30)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(31)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(32)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − in(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(33)

Tst(j, jst, n + 1) ≥ Tst(j, jst, n) − UH(1 − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(34)

As we mentioned in Section 2, there are two types of operating scenario for storage tanks and for each one of them, we have the following additional constraints.

Scenario 1: Simultaneous loading and unloading not allowed

Scenario 1 represents that material cannot flow in and out of the storage tank at the same time at any event point (n). This operation rule is used in many refineries for security reasons. Constraints (36) represent that output flow from storage tank (jst) starts after input ends at any event point (n)

Tst(j, jst, n) − UH(1 − in(j, jst, n)) ≤ Tst(j, jst, n) + UH(1 − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(36)

Scenario 2: Simultaneous loading and unloading allowed

The operation rule for scenario 2 is that material can flow in and out of tank at the same time at any time interval (n). This assumption is very common in many refineries for intermediate storage tanks.

Constraints (37)–(40) are enforced when both variables in(jst,n) and out(jst,n) are equal to 1. When the constraints are enforced, the starting and finishing times of loading and unloading events are equal. When either in(jst,n) or out(jst,n) is equal to zero, the constraints are relaxed.

Tst(j, jst, n) + UH(1 − in(j, jst, n))
≥ Tst(j, jst, n) − UH(1 − out(j, jst, n)),
∀ jst ∈ Jst, j ∈ Jprodst(jst), n ∈ N
(37)
Tss(j, jst, n) – UH(1 − in(j, jst, n))
\leq Tss(jst, j, n) + UH(1 − out(jst, j, n)),
∀jst ∈ Jst, j ∈ Jprod(jst), jst, j, n ∈ N

(38)

|TSF(j, jst, n) + UH(1 − in(j, jst, n))| ≥ TSF(jst, j, n) – UH(1 − out(jst, j, n)),
∀jst ∈ Jst, j ∈ Jprod(jst), jst, j, n ∈ N

(39)

3.1.8. Makespan constraints

Constraints (41)–(46) state that starting and finishing time of any task is always less than equal to the makespan (H).

Ts(i, j, n) ≤ H, ∀i ∈ I, j ∈ J(i), n ∈ N

(41)

Tf(i, j, n) ≤ H, ∀i ∈ I, j ∈ J(i), n ∈ N

(42)

Tss(i, jst, n) ≤ H, ∀jst ∈ Jst, j ∈ Jprod(jst), n ∈ N

(43)

TSS(j, jst, n) ≤ H, ∀jst ∈ Jst, j ∈ Jprod(jst), n ∈ N

(44)

Tsf(j, jst, n) ≤ H, ∀jst ∈ Jst, j ∈ Jprod(jst), n ∈ N

(45)

Tsf(jst, j, n) ≤ H, ∀jst ∈ Jst, j ∈ Jprod(jst), n ∈ N

(46)

3.1.9. Objective function

Finally, Eq. (47) defines the objective function of the problem which is the minimization of makespan. The most common motivation for optimizing the process using minimization of makespan as objective function is to improve customer services by accurately predicting order delivery dates. Moreover, when specific due dates are not available, minimization of makespan is an appropriate objective function (Maravelias, 2006; Ierapetritou & Floudas, 1998).

z = \min(H)

(47)

4. Solution approach

The CPU time for the solution of the overall model presented in the previous section is usually high due to model size for large scale scheduling problems (Bassett et al., 1996; Roslof, Harjunkoski, Westerlund, & Isaksson, 2002). In order to reduce the CPU resolution time we developed a structural decomposition strategy which decomposes the problem into a number of smaller and thus easier to solve sub-systems. The developed structural decomposition approach and the additional constraints presented in this section guarantee that the solution obtained by the decentralized optimization will be feasible for the centralized system and is exactly the same.

4.1. Structural decomposition approach

Generally, scheduling problems are large scale problems and are difficult to be solved to optimality. As the scale of the production increases, the mathematical complexity of the developed model increases, and the CPU time that is required for the solution of the corresponding problem increases too. Decomposition is a natural way to deal with large scale problems. There are two types of decomposition: the structural decomposition of the system under study and the mathematical decomposition such as Benders decomposition (Benders, 1962), Lagrangian relaxation (Minoux, 1986), etc. Note that mathematical decomposition can be applied after the application of the structural decomposition if it is applicable.

The decomposition strategy proposed here decomposes the refinery scheduling problem presented in Section 2 spatially. To obtain the optimal solution in decentralized optimization approach, each sub-system is solved to optimality and these optimal results are then used to obtain the optimal solution for the entire problem. In our proposed decomposition rule, we split the system in such a way that the numbers of constraints that are needed in each sub-system to ensure feasibility of the overall system are as few as possible. Furthermore, our aim is to create a decentralized problem that needs minimum number of iterations between sub-systems to obtain a feasible solution for the global problem. To create such a decentralized system, we propose a decomposition rule as follows.

We split the problem at intermediate storage tanks such that inlet and outlet streams of a tank belong to different sub-systems. The decomposition starts with the final products or product storage tanks, and continues to include the production units that are connected to them and stops when the storage tanks are reached. The products, intermediate products, units and storage tanks are part of the sub-system 1. Then following the input stream of each storage tank, the same procedure is used to determine the next sub-system. If inlet and outlet streams of the tank are included in the same local problem then the storage tank also belongs to that local problem. Fig. 2 presents a decomposition of the refinery production problem presented in Section 2 after application of the proposed decomposition rule.

In decentralized system, the individual sub-systems are treated as an independent scheduling problem with their own products and raw materials. The sub-system k receives raw materials from sub-system k + 1. These individual sub-systems are solved to optimality using the mathematical formulation described in Section 3 that follows the same operating rules as those required for the centralized problem. In decentralized system, the intermediate storage tanks (where the system is decomposed) are becoming the raw material tanks for the sub-problem k in decentralized problem. We first solve sub-system 1 to optimality, which gives us the information about the raw materials' demand for the connecting storage tanks in sub-problem 1. This information is passed to sub-system 2 which is then solved to the optimality using these data. Thus if the decentralized system is decomposed into K sub-systems, we first solve sub-system 1 and sub-system K last. To obtain the solution for the global problem in the decentralized solution approach, we integrate the optimal solution of each sub-system such that the mass balance and operating constraints for each connecting storage tank are satisfied. Section 4.2 presents the constraints necessary for obtaining a feasible solution for the global problem in decentralized solution approach. These additional constraints are used in the decentralized approach in order to guarantee that the solution obtained by the sub-system k could be realized by the sub-system k + 1.

4.2. Constraints for decentralized model

The sub-systems obtained using the decomposition rule presented in the previous section, have all the constraints presented in the basic model. Only few additional constraints are required to connect the sub-systems via intermediate products' demand and to enforce feasibility of the global problem solution. The decomposition is done so that the sub-systems interact with each other only via the connecting storage tanks. Thus, we obtain K independent sub-systems that have the same characteristics as the global system. This choice of boundary requires minimum information to be shared between the sub-systems in the decentralized approach. Moreover, there is no need to impose any additional sequence and due date constraints in decentralized model.
4.2.1. Demand constraints

In decentralized system, the $k+1$ sub-system can produce two types of final products: the finished final products present in global system; and the products required by sub-system $k$ as a raw materials, which are defined as intermediate products in global system. The demand constraints for the first type of products are presented as part of the basic model in Section 3. However, we need additional constraints in decentralized model for the second type of products.

The basic mathematical model presented in Section 3 is based on continuous time representation using event points Ierapetritou et al. (1999). In decentralized system, after solving sub-system on continuous time representation using event points, the amount needed for each connecting storage tank at each event, we obtain the information about the material type and amount required by the sub-system to optimality. The demand constraints for intermediate final products given by Eq. (48), where the parameter is determined by solving the sub-system $k$ to optimality.

$$\sum_{j} \text{outflow}2(s, j, n) \geq r(s, n), \quad \forall s \in S, j \in \text{inj}(s, k + 1), n \in N$$

(48)

Scenario 1: Simultaneous loading and unloading is not allowed.

4.2.2. Feasibility constraints

Production units in sub-system $k+1$ supply material necessary for production to sub-system $k$ in decentralized system. The demand constraint for intermediate final products given by Eq. (48) allows sub-system $k+1$ to produce more products than the amount required by the sub-system $k$. Thus the amount of intermediate products supplied to connecting storage tanks by sub-system $k+1$ should be limited by the storage capacity of the tanks so that integrated solution of the decentralized system is feasible for the original global problem. Thus when production units in sub-system $k+1$ supply material to storage tanks located in sub-system $k$, in order to obtain globally feasible solution, the following capacity constraints are added to sub-system $k+1$.

$$\sum_{j} \text{outflow}2(s, j, n) + \text{stin}(jst) \leq V_{\text{max}}(jst), \quad \forall s \in S, jst \in S_j(s, k).$$

(49)

$$\sum_{j} \text{outflow}2(s, j, n) + \text{stin}(jst) - r(s, 0) \leq V_{\text{max}}(jst), \quad \forall s \in S, jst \in S_j(s, k), n \in N$$

(50)

$$\sum_{j} \text{outflow}2(s, j, n) + \text{stin}(jst) - \sum_{n=0}^{m-1} r(s, n) \leq V_{\text{max}}(jst), \quad \forall s \in S, jst \in S_j(s, k), n \in N$$

(51)

Constraints (49) are enforced for event point $n=0$ and express the requirement that the summation of the material supplied to storage tank $(jst)$ in sub-system $k$ and initial amount present in the storage tank $(\text{stin}(jst))$ must not exceed the tank capacity $(V_{\text{max}}(jst))$. Constraints (50) represent the capacity constraints for event point $n=1$. For event point $n=m$, the feasible constraint is given by Eq. (51). The total number of feasibility constraints present in sub-system $k+1$ is equal to the total number of event points utilized by sub-system $k$ in obtaining the optimal solution.

4.2.3. Equilibrium constraints

Since as stated previously, sub-system $k+1$ can produce more than the required intermediate products by sub-system $k$, when the production in sub-system $k+1$ exceeds the set demand limit in event point $n$, sub-system $k+1$ should produce less than necessary during event point $n+1$. Thus, additional constraints are necessary to be added to decentralized sub-system $k+1$ so that the lot-sizing characteristic of global problem can be captured.

$$r(s, 1) - \left( \sum_{j} \text{outflow}2(s, j, 0) + \text{stin}(jst) - r(s, 0) \right) = r(s, 1), \quad \forall s \in S, j \in \text{inj}(s, k + 1), jst \in S_j(s, k)$$

(52)

$$r(s, 2) - \left( \sum_{j} \sum_{n=0}^{1} \text{outflow}2(s, j, n) + \text{stin}(jst) - \sum_{n=0}^{1} r(s, n) \right) = r(s, 2), \quad \forall s \in S, j \in \text{inj}(s, k + 1), jst \in S_j(s, k), n \in N$$

(53)
equal to the maximum rate of consumption in sub-system $k$ times duration of production in sub-system $k+1$. Constraints (55) and (58) together implement the operating rule of simultaneous loading and unloading and mass balance constraints for connecting storage tanks. Furthermore, these two constraints enforce similar makespan for all sub-systems when the objective function is minimization of makespan.

As stated previously, sub-systems other than sub-system 1 can produce finished final products for global problem. In this case, the optimal makespan for the sub-system $k+1$ can be larger than that of sub-system $k$ so we have to resolve sub-system $k$ to optimality with constraint (59) active in the model.

The makespan of each sub-system will be the same in scenario 2 when additional constraints (55)–(59) are present and minimization of makespan is used as an objective function. Furthermore, the global optimal solution is obtained by combining the optimal schedule of each sub-system. Constraints (55), (58) and (59) ensure the satisfaction of the material balance requirements for intermediate connecting tanks. If one of the sub-systems is infeasible, then the global system is infeasible too since the additional feasibility constraints in scenarios 1 and 2 describe the solution space of the global problem where the primary finished products are produced by sub-system 1.

The solution obtained from the decentralized system is the same as the one for the original problem. This is due to the following reasons. First the way the system is decomposed, at the intermediate storage tanks, the sub-systems interact with each other only through the connecting storage tanks. Second we assume fixed recipe and thus under variable production rate for each unit the minimum makespan is achieved only when all the production units operate at the maximum feasible production rate. This highest feasible production rate of unit ($j$) is calculated by considering the units that come before and after unit ($j$). Thus, the highest feasible production rate for unit ($j$) depends only on the parameters and the configuration of the system and is thus the same in centralized and decentralized systems. Furthermore, all the interactions between sub-systems are captured by additional constraints that are shown in Section 4. The solution of the centralized and decentralized problems considering different objective functions such as minimization of cost is considered in our current work and will be the subject of future publication.

5. Numerical results

To illustrate the applicability and effectiveness of the decentralization strategy proposed in this paper, a refinery production scheduling case study based on realistic data provided by Honeywell Hi-Spec Solutions is presented in this section. The details of the refinery are shown in Fig. 4.

In this system the production starts from crude oil distillation units and proceed to diesel blender unit to produce home heating oil (Red Dye diesel) and automotive diesel (Carb diesel and EPA diesel). Crude distillation unit, 4CU, processes Alaskan North Slope (ANS) crude oil which is stored in raw material storage tanks ANS1 and

\[
r(s, m) - \left( \sum_{j=0}^{m-1} \sum_{n=0}^{j} \text{outflow2}(s, j, n) + s \text{tin}(st) - \sum_{n=0}^{j} r(s, n) \right) = r(s, m), \quad \forall s \in S, j \in \text{Junit}(s, k+1), n \in N
\]  

(54) Constraints (52) and (53) represent equilibrium constraints for sub-system $k+1$ for event point $n=1$ and $n=2$ respectively. For event point $n=m$, equilibrium constraints is given by Eq. (54). The demand of intermediate final product $s$ at event point $n$ is adjusted by the amount present in the storage tank after the demand is satisfied at previous event point ($n-1$). This adjusted demand $r(s, n)$ is then used in demand constraints for intermediate final products instead of using the original demand $r(s, n)$. The total number of equilibrium constraints present in sub-system $k+1$ is equal to $(N-1)$, where $N$ is the total number of event points utilized by sub-system $k$ in obtaining the optimal solution.

The optimal time horizon of global problem is obtained by combining the optimal schedules of sub-systems at each point ($n$) such that the material balance constraints are satisfied for connecting intermediate storage tanks. Since sub-system $k+1$ satisfies the demand of sub-system $k$, sub-system $k+1$ will happen before the sub-system $k$ at each event points. For refinery production system presented in Section 2, we assume that all finished primary final products are produced by sub-system 1 and this assumption is common for many refinery systems. Since the demand for primary final products are much higher than the demands for secondary final products, the demands of primary products will play a significant role in determining the production makespan.

Scenario 2: Simultaneous loading and unloading allowed.

The objective function is to minimize the makespan and the only way this objective can be realized is by operating all units at their maximum feasible production rate. When we decompose the original system at intermediate storage tanks, the interactions between all production units in each sub-system is maintained and is the same as in the global system. Thus, the maximum feasible production rate in each sub-system is known a priori. To implement the constraint of simultaneous loading and unloading to intermediate connecting storage tanks between any two sub-systems $k$ and $k+1$, the following constraints are added to sub-system $k$.

\[
\text{inflow1}(jst, n) = \beta(s) \times (Ts(jst, j, n) − Tss(jst, j, n)),
\]  

\[
\forall s \in S, jst \in \text{Jst}(s, k), j \in \text{Junit}(s, k), n \in N
\]  

(55) \[
\beta(s) = \frac{1}{\text{max}(\text{max}(\text{outflow1}(s, j, n))), \text{outflow2}(s, j, n)}\]  

\[
\forall s \in S
\]  

(56) \[
\beta(s) = \text{max}(\text{max}(\text{inflow1}(s, j, n)), \text{outflow2}(s, j, n))\]  

\[
\forall s \in S
\]  

(57) where $\text{outflow1}(s, j, n)$ is the maximum rate of production of intermediate final product $s$ in sub-system $k$ and $\text{outflow2}(s, j, n)$ is the maximum rate of consumption of $s$ in sub-system $k$. $\text{outflow1}(s, j, n)$ and $\text{outflow2}(s, j, n)$ can be calculated before the start of the optimization process based on the configuration of sub-systems. $\beta(s)$ takes the value of either the maximum rate of production or maximum rate of consumption as given by constraints (56) and (57). Constraint (55) determines the amount of material inflow to the storage tank at event point ($n$). When $\beta(s)$ takes the value of maximum rate of consumption, constraint (58) is added to the sub-system $k+1$.

\[
\sum_{j} \text{outflow2}(s, j, n) = \beta(s) \times (Ts(i, j, n) − Tsl(i, j, n)),
\]  

\[
\forall s \in S, j \in \text{Junit}(s, k+1), i \in I(j), n \in N
\]  

(58) Constraints (58) state that the total amount of material $s$ which is an intermediate final product, produced in sub-system $k+1$ is

As stated previously, sub-systems other than sub-system 1 can produce finished final products for global problem. In this case, the optimal makespan for the sub-system $k+1$ can be larger than that of sub-system $k$ so we have to resolve sub-system $k$ to optimality with constraint (59) active in the model.

\[
\text{inflow1}(jst, n) = \text{iter} \times \sum_{j} \text{outflow2}(s, j, n), \forall s \in S, jst \in \text{Jst}(s, k),
\]  

\[
j \in \text{Junit}(s, k+1), n \in N
\]  

(59) where $\text{iter}$ is a binary variable which is equal to 1 if the optimal makespan of sub-system $k+1$ is greater than sub-system $k$, and 0 otherwise. $\text{outflow2}(s, j, n)$ is a parameter which is obtained from the optimal solution of sub-system $k+1$. Constraints (59) specify the amount of material flowing into the connecting intermediate storage tank in sub-system $k$ from sub-system $k+1$ at each event point ($n$).

The makespan of each sub-system will be the same in scenario 2 when additional constraints (55)–(59) are present and minimization of makespan is used as an objective function. Furthermore, the global optimal solution is obtained by combining the optimal schedule of each sub-system. Constraints (55), (58) and (59) ensure the satisfaction of the material balance requirements for intermediate connecting tanks. If one of the sub-systems is infeasible, then the global system is infeasible too since the additional feasibility constraints in scenarios 1 and 2 describe the solution space of the global problem where the primary finished products are produced by sub-system 1.
ANS2, whereas crude distillation unit 2 (2CU) processes San Joaquin Valley (SV) crude oil. SV crude oil is supplied to 2CU via pipeline. The products of crude distillation units are then processed further downstream by vacuum distillation tower unit and diesel high pressure desulfurization (HDS) unit. The coker unit converts vacuum resid into light and heavy gasoil and produces coke as residual product. The fluid catalyzed high pressure desulfurization (FCC HDS) unit, FCC, Isomax unit produce products that are needed for diesel blender unit. The FCC unit also produces by-product FCC gas. The diesel blender blends HDS diesel, hydro diesel, and light cycle oil (LCO) to produce three different final products. The diesel blender sends final products to final product storage tanks. The by-product FCC gas and residual product Coke are not stored but supplied to the market via pipeline. The system employs four storage tanks to store intermediate products, vacuum resid, diesel, light gasoil, and heavy gasoil.

In Fig. 4 we present the decomposition of the system under study after the application of the developed decomposition rule. The system is split in two sub-systems where sub-system 1 produces all of the final products and one by-product. The sub-system 1 includes 5 production units, 7 final product storage tanks, and 3 raw material tanks. Raw material tanks in sub-system 1 are defined as intermediate tanks in centralized system. The sub-system 2 includes 4 production units, 1 intermediate tank, 2 raw material tanks and it produces 4 final products. Except Coke, all other final products in sub-system 2 are defined as intermediate products in centralized system.

The data for the problem studied here are presented in Appendix A. Different demand cases for final products, Carb diesel, EPA diesel, and Red Dye diesel; and residual products, Coke and FCC gas, are studied. The actual values of the products' demands are given in Table A1 (Appendix A). For all computations in this paper GAMS/CPLEX 10.0 is used to solve the resulting MILP formulations. The optimal solution is obtained with 1e−6 integrality gap using a Pentium(R) 4 processor at 3.40 Hz and with 1.99GB memory. The two scenarios presented in Section 3, with and without the assumption of simultaneously loading and unloading of tank are examined. Centralized and decentralized optimization is applied using the decomposition approach presented in Section 4. In the following tables four different examples are presented in order to illustrate the advantage of decentralized optimization using the decomposition approaches. Four examples represent lower to high demands for the system that need to be satisfied within available time horizon of 240 h.

Following the mathematical model presented is Section 3, the 1st sub-system as shown in Fig. 4 is solved to minimize the makespan and meet the demand for the final products. Based on the optimal solution of sub-system 1, sub-system 2 is solved to optimality such that it satisfies the demand required by sub-system 1. In the end the solutions of the two sub-systems are combined to obtain the solution of the entire problem. The computational characteristics of the problems for scenarios 1 and 2 are shown in Tables 1–4.

As shown in Tables 1 and 2, after the application of the decomposition strategy the size of sub-systems is significantly reduced (ex. scenario 1 form 1081 to 749) giving rise to a small increase in the number of constraints. This happens because the decision variables associated with connecting the two sub-systems in centralized problem become demand data in decentralized approach resulting to additional constraint to the first sub-system.

To obtain the global optimal solution for scenario 1, the optimal schedules of each sub-system are combined at each event point n such that the material balance and storage capacity constraints for intermediate connecting tanks are satisfied, whereas the global optimal schedule in scenario 2 is obtained by superimposing the optimal solution of each sub-system. As shown in Tables 3 and 4 for both scenarios, the centralized and decentralized optimizations give exactly the same optimal makespan for all examples. As explained in the following section, sub-system 1 has exactly the same solution in centralized and decentralized optimization which gives rise to the same optimal solution (centralized and decentralized) for sub-system 2.

The objective function in centralized and decentralized strategy is minimization of makespan. In order to spend minimum time producing material (s), it is required to operate all the units in the system in such a way that they will produce all materials needed at a maximum production rate that is feasible for that particular system. The maximum possible production rate \( P_{\text{max}}(s) \) for each product (s) can be determined based on the data given in Appendix A. In sub-system 2, only one unit (j) produces the specific material (s), which means that \( P_{\text{max}}^{s+1}(s) \) is defined only by the highest feasible production rate of the unit (j). Since, our proposed decom-

---

**Table 1**

<table>
<thead>
<tr>
<th></th>
<th>Continuous Variables</th>
<th>Binary Variables</th>
<th>Total number of variables</th>
<th>Constraints</th>
<th>Total number of constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>985</td>
<td>96</td>
<td>1081</td>
<td>1425</td>
<td>1425</td>
</tr>
<tr>
<td>Sub System 1</td>
<td>484</td>
<td>54</td>
<td>749</td>
<td>921</td>
<td>1686</td>
</tr>
<tr>
<td>Sub System 2</td>
<td>187</td>
<td>24</td>
<td>765</td>
<td>765</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th></th>
<th>Continuous Variables</th>
<th>Binary Variables</th>
<th>Total number of variables</th>
<th>Constraints</th>
<th>Total number of constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>985</td>
<td>96</td>
<td>1081</td>
<td>1488</td>
<td>1425</td>
</tr>
<tr>
<td>Sub System 1</td>
<td>484</td>
<td>54</td>
<td>743</td>
<td>930</td>
<td>1485</td>
</tr>
<tr>
<td>Sub System 2</td>
<td>181</td>
<td>24</td>
<td>555</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3  Scenario 1: Centralized vs. decentralized.

<table>
<thead>
<tr>
<th>Example</th>
<th>Centralized System</th>
<th>Decentralized System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU Time (s)</td>
<td>Objective Value (h)</td>
</tr>
<tr>
<td>1</td>
<td>318.672</td>
<td>235.890</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>256.187</td>
<td>163.352</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2192.515</td>
<td>79.328</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>589.984</td>
<td>26.149</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4  Scenario 2: Centralized vs. decentralized.

<table>
<thead>
<tr>
<th>Example</th>
<th>Centralized System</th>
<th>Decentralized System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU Time (s)</td>
<td>Objective Value (h)</td>
</tr>
<tr>
<td>1</td>
<td>4.125</td>
<td>235.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5.844</td>
<td>215.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5.562</td>
<td>191.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7.562</td>
<td>97.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

position rule does not affect the interactions and configuration of the productions units that are directly connected to each other, the maximum feasible production rate of each sub-system is the same in centralized and decentralized systems. Furthermore, additional constraints added in the decentralized model capture all the interactions between sub-systems due to the existence of storage tanks. Thus, the same optimal solution is obtained with both approaches.

As shown in Tables 3 and 4, the CPU time required to find the optimal makespan is reduced significantly in decentralized approach compared to centralized approach. This is mainly due to the reduction in size and complexity of the system going from centralized system to sub-systems.

For scenario 1, the CPU time needed to solve the centralized system is in the order of 100 s whereas the decentralized system is solved within 5 CPU seconds. In scenario 2, decentralized solution approach also show improvement compared to centralized system. The CPU time needed to solve the problem is cut by half in decentralized system compared to centralized system. For scenario 1, the production schedule obtained by decentralized approach is different than that obtained by centralized approach as shown in Figs. 5 and 6 for example 1. This difference in production schedule is obtained because in decentralized system, an optimal solution is obtained by integrating the schedules of each sub-system at each event point. Gantt charts for storage tanks are given in Appendix A for centralized and decentralized systems and for scenario 2.

6. Summary and future directions

In this paper, a structure decomposition strategy and formulation is presented for short-term scheduling of refinery operations. It is shown that the decentralized system model results in fewer constraints and fewer continuous and binary variables compared...
to centralized system. The paper presents a problem where both optimization strategies result in the same optimal makespan but the computational time for decentralized system is reduced significantly compared to that of centralized system. Currently we are investigating the system but with variable production recipe which gives rise to nonlinear equations in the mathematical formulation. Another direction is to examine the solutions given from centralized and decentralized strategy under different objective functions, such as maximization of profit, minimization of the inventory in the tanks.

Acknowledgments

The authors would like to thanks Jeff Kelly from Honeywell for providing the case study presented in the paper. The authors gratefully acknowledge financial support from the National Science Foundation under the Grant CTS 0625515.

Appendix A.


### Table A1
Demand in Thousand barrels.

<table>
<thead>
<tr>
<th>Final Products</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
<td>Case 2</td>
</tr>
<tr>
<td>FCC gas</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Coke</td>
<td>10</td>
<td>35</td>
</tr>
<tr>
<td>Carb diesel</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>EPA diesel</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Red dye diesel</td>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table A2
Production rates in thousand barrels/hour.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Task</th>
<th>$r_{\text{min}}$</th>
<th>$r_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CU</td>
<td>4CU Normal</td>
<td>0.1</td>
<td>7.292</td>
</tr>
<tr>
<td>2CU</td>
<td>2CU Normal</td>
<td>0.1</td>
<td>4.167</td>
</tr>
<tr>
<td>Vacuum Tower</td>
<td>Vacuum Normal</td>
<td>0.1</td>
<td>5.708</td>
</tr>
<tr>
<td>Coker</td>
<td>Coker Normal</td>
<td>0.1</td>
<td>2.75</td>
</tr>
<tr>
<td>FCC HDS</td>
<td>FCCHDS Normal</td>
<td>0.1</td>
<td>3.00</td>
</tr>
<tr>
<td>FCC</td>
<td>Maxdistillation Mode</td>
<td>0.1</td>
<td>2.708</td>
</tr>
<tr>
<td>Diesel</td>
<td>Diesel Normal</td>
<td>0.1</td>
<td>2.708</td>
</tr>
<tr>
<td>Isomax</td>
<td>Isomax Normal</td>
<td>0.1</td>
<td>2.5</td>
</tr>
<tr>
<td>Diesel Blender</td>
<td>Carb Diesel</td>
<td>0.1</td>
<td>2.042</td>
</tr>
<tr>
<td>Diesel Blender</td>
<td>EPA Diesel</td>
<td>0.1</td>
<td>54.167</td>
</tr>
<tr>
<td>Diesel Blender</td>
<td>RedDye Diesel</td>
<td>0.1</td>
<td>54.167</td>
</tr>
</tbody>
</table>

### Table A3
Recipe data.

<table>
<thead>
<tr>
<th>Task</th>
<th>State produced</th>
<th>$\rho^{q}$</th>
<th>State consumed</th>
<th>$\rho^{c}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CU Normal</td>
<td>Diesel</td>
<td>0.691</td>
<td>ANS1</td>
<td>1</td>
</tr>
<tr>
<td>Normal</td>
<td>Resid</td>
<td>0.309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2CU Normal</td>
<td>Diesel</td>
<td>0.284</td>
<td>SJV Crude</td>
<td>1</td>
</tr>
<tr>
<td>Normal</td>
<td>Resid</td>
<td>0.716</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vacuum Tower</td>
<td>Vacuum Resid</td>
<td>0.334</td>
<td>Resid</td>
<td>1</td>
</tr>
<tr>
<td>Normal</td>
<td>Heavy Gasoil</td>
<td>0.333</td>
<td>Light Gasoil</td>
<td>1</td>
</tr>
<tr>
<td>Coker</td>
<td>Coke</td>
<td>0.640</td>
<td>Vacuum Resid</td>
<td>1</td>
</tr>
<tr>
<td>Normal</td>
<td>Heavy Gasoil</td>
<td>0.180</td>
<td>Light Gasoil</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCC HDS Normal</td>
<td>FCC HDS</td>
<td>1</td>
<td>Heavy Gasoil</td>
<td>1</td>
</tr>
<tr>
<td>Maxdistillation Mode</td>
<td>FCC Gas</td>
<td>0.50</td>
<td>FCC HDS</td>
<td>1</td>
</tr>
<tr>
<td>Maxgasoline Mode</td>
<td>FCC Gas LCO</td>
<td>0.50</td>
<td>FCC HDS</td>
<td>1</td>
</tr>
<tr>
<td>Diesel Normal</td>
<td>HDS Diesel</td>
<td>1</td>
<td>Diesel</td>
<td>1</td>
</tr>
<tr>
<td>Isomax Normal</td>
<td>Hydro Diesel</td>
<td>1</td>
<td>Light Gasoil</td>
<td>1</td>
</tr>
<tr>
<td>Carb Diesel</td>
<td>Carb Diesel</td>
<td>1</td>
<td>HDS Diesel</td>
<td>0.20</td>
</tr>
<tr>
<td>Normal</td>
<td>Carb Diesel</td>
<td>1</td>
<td>Hydro Diesel</td>
<td>0.25</td>
</tr>
<tr>
<td>EPA Diesel</td>
<td>EPA Diesel</td>
<td>1</td>
<td>HDS Diesel</td>
<td>0.25</td>
</tr>
<tr>
<td>Normal</td>
<td>EPA Diesel</td>
<td>1</td>
<td>Hydro Diesel</td>
<td>0.15</td>
</tr>
<tr>
<td>Red Dye Diesel</td>
<td>Red Dye Diesel</td>
<td>1</td>
<td>HDS Diesel</td>
<td>0.23</td>
</tr>
<tr>
<td>Normal</td>
<td>Red Dye Diesel</td>
<td>1</td>
<td>Hydro Diesel</td>
<td>0.12</td>
</tr>
</tbody>
</table>

### Table A4
Storage tank capacity data in thousand barrels.

<table>
<thead>
<tr>
<th>Storage Tank</th>
<th>Material Stored</th>
<th>Capacity</th>
<th>Initial Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANS Feed Tank 1</td>
<td>ANS</td>
<td>750</td>
<td>10</td>
</tr>
<tr>
<td>ANS Feed Tank 2</td>
<td>ANS</td>
<td>750</td>
<td>10</td>
</tr>
<tr>
<td>Coker Feed Tank</td>
<td>Vacuum Resid</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>FCC HDS Feed Tank</td>
<td>Heavy Gasoil</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>Diesel HDS Feed Tank</td>
<td>Diesel</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>Isomax Feed Tank</td>
<td>Light Gasoil</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>Carb Diesel Tank 1</td>
<td>Carb Diesel</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Carb Diesel Tank 2</td>
<td>Carb Diesel</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Carb Diesel Tank 3</td>
<td>Carb Diesel</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Carb Diesel Tank 4</td>
<td>Carb Diesel</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>EPA Feed Tank 1</td>
<td>EPA Diesel</td>
<td>125</td>
<td>0</td>
</tr>
<tr>
<td>EPA Feed Tank 2</td>
<td>EPA Diesel</td>
<td>125</td>
<td>0</td>
</tr>
<tr>
<td>RedDye Diesel Tank 1</td>
<td>RedDye Diesel</td>
<td>150</td>
<td>0</td>
</tr>
<tr>
<td>RedDye Diesel Tank 2</td>
<td>RedDye Diesel</td>
<td>150</td>
<td>0</td>
</tr>
</tbody>
</table>
Fig. A1. Gantt chart of tank loading schedule for example 1, centralized system, scenario 1.

Fig. A2. Gantt chart of the tank unloading schedule for example 1, centralized system, scenario 1.

Fig. A3. Gantt chart of tank loading schedule for example 1, decentralized system, scenario 1.
Fig. A4. Gantt chart of the tank unloading schedule for example 1, decentralized system, scenario 1.

Fig. A5. Gantt chart of the operation schedule for example 1, centralized system, scenario 2.

Fig. A6. Gantt chart of the operation schedule for example 1, decentralized system, scenario 2.
Fig. A7. Gantt chart of tank loading schedule for example 1, centralized system, scenario 2.

Fig. A8. Gantt chart of tank unloading schedule for example 1, centralized system, scenario 2.

Fig. A9. Gantt chart of tank loading schedule for example 1, decentralized system, scenario 2.
Fig. A10. Gantt chart of tank unloading schedule for example 1, decentralized system, scenario 2.

References


