Optimum Scheduling of Energy Production Using Hybrid Decomposition Methods – A Comparative Study

التخطيط الأمثل لإنتاج الطاقة بإستخدام طرق تحليل ثنائية – دراسة مقارنة

Alsaied Khalil, Dr. Eng.
Member ASHRAE

Department of Mechanical Power Engineering, Faculty of Engineering, Tanta University, Tanta, Egypt, E-mail: akhalileg@yahoo.com

ملخص

تتواجد الطاقة فى صور مختلفة ويمكن الحصول عليها من مصادر مختلفة. يعتبر تخطيط وإنتاج أنواع الطاقة المختلفة عملية إتخاذ قرار بتخصيص المصادر الطبيعية والصناعية بأفضل ظروف إقتصادية. ويمثل إنتاج الطاقة تحديا كبيرا بسبب إزدياد الطلب عليها وتغير تكاليف إنتاجها وعدم القدرة على التنبؤ الدقيق بالطلب عليها فى المستقبل. تستخدم طريقة أحادية المستوى لحل هذه المشكلة حيث تعالجها كمشكلة برمجة خطية مختلطة ذات حجم كبير. أما طريقة التحليل ذات المستويات المتراتبة فتقوم بتقسيم المشكلة إلى مستويين. وتتميز طريقة تحليل المستويات بأنها تعالج مسألة عدم الدقة فى البيانات بطريقة أفضل، كما أنها تتطلب جهدا حسابيا أقل.

فى هذا البحث يتم المقارنة بين طريقتين من طرق تحليل المستويات المتراتبة لمشكلة تخطيط وانتاج الطاقة. تعتمد الطريقتان على تقسيم المشكلة الرئيسية إلى مشكلتين فرعيتين: الأولى فى مستوى منخفض يحتوى على مجموعة من الأسر والتى تتجمع بدورها معا فى مستوى مرتفع لتكون مجموعة من الأنماط الرئيسية للطاقة. فى **الطريقة الأولى** يتم إدماج مجموعة قيود مستودعات التخزين فى دالة الهدف لنحصل بعد تقسيم المشكلة الرئيسية على مشكلة فرعية على المستوى المنخفض (مشكلة تحكم أمثل) وأخرى على المستوى العلوى عبارة عن مشكلة برمجة خطية. أما فى **الطريقة الثانية** فيتم تقسيم النموذج أولا، ثم دمج مجموعة قيود فى دالة الهدف للمشكلة الفرعية (مستوى منخفض). ينتج عن ذلك مشكلة فرعية بسيطة ذات حل واضح نحصل علية بالفحص المباشر، بالإضافة إلى مشكلة فرعية اخرى مشابهة لمشكلة التحكم الأمثل فى الطريقة الأولى يتم حلها فى الطريقتين بإستخدام البرمجة الديناميكية. تم إجراء تبسيط ملحوظ فى الطريقة الثانية كما تم إيجاد وسيلة عالية الكفاءة للتحكم فى حجم خطوة التقارب العددى. تم بناء مجموعة برامج حاسب آلى بلغة الفورتران للحصول على الحلول. توصلت النتائج النهائية لعدد ٣٦٠ مشكلة ذات مدخلات مختلفة على أن كلا الطريقتين تتميز بكفاءة مقبولة. كما أظهرت النتائج حدوث تحسن ملحوظ فى أداء الطريقتين. ويعزى السبب فى هذا التحسن إلى التحكم فى حجم خطوة التقارب العددى وإلى التبسيط الذى تم إدخاله على طرق الحل.

ABSTRACT

The production of energy exists in different shapes and forms, e.g. thermal, mechanical, electrical or other forms of energy. Each of these forms exists in different types and arts. The production and scheduling of energy types are decision processes which are basically concerned with the adoption or more precisely the allocation of the natural and industrial resources in order to best satisfy marketing and customer requirements at minimum best economic conditions. There are many reasons that make energy production planning a challenging problem. The variable structure of both the demand and costs, the difficulty to precisely forecast the demand at the very detailed level, the less flexibility to modify the operating conditions, and that there are usually long and uncertain delays in obtaining the industrial resources (new machines, workers training, subcontracting capacities, raw materials، item components, etc.)

Researchers and scientists have paid a great part of attention for developing scheduling and planning systems to support such decision making processes. Two different approaches for production planning and scheduling are known. The first, called the *monolithic approach*, formulates the problems as a large scale mixed-integer linear programming problems and is usually solved approximately using Lagrangean relaxation to the mixed integer linear program. The second approach, called the *hierarchical approach*, partitions the planning and scheduling problem into a hierarchy of smaller subproblems. The upper hierarchy deals with strategic decisions for the planning horizon, while the lower hierarchy deals with the more detailed short-termed scheduling.

Accepted July 26, 2004.

The computational effort required for the monolithic approach is generally greater than that required for the hierarchical approach. The hierarchical approach may require less detailed demand input data. Also the uncertainty is better treated by the hierarchical approach.

In this study, a comparison is presented between two different hybrid procedures for solving the hierarchical energy production planning and scheduling problem. A simplification in one of them is made. In the first procedure, hierarchical subproblems are included in an overall mixed-integer linear programming formulation. Accordingly, the problem is partitioned into two hierarchies; families in the lower hierarchy, which, in turn, are aggregated into types in the upper hierarchy. In this procedure, the set of inventory constraints are priced out using Lagrangean multipliers in the main objective function, which is then partitioned into two subproblems. The first subproblem is an optimal control problem on the family level (lower hierarchy) while the other subproblem is a linear program on the type level (upper hierarchy). The two subproblems are linked together by an inventory aggregation constraint. By this approach, a feedback is included automatically in the solution procedure, but it does not pass directly from the lower level to the upper level since both subproblems are solved in parallel. In the second procedure, instead of using Lagrangean relaxation with respect to a group of constraints in the original model, the model is first partitioned, in which cuts are priced out by means of a set of Lagrangean multipliers into the subproblem generated on the family level. Accordingly, one subproblem is trivial and the other subproblem is an incapacitated lot-sizing optimal control problem. The size of this subproblem is exactly the same as that of the equivalent subproblem in the previous procedure. In both procedures, the optimal control subproblem is solved using dynamic programming and the set of Lagrangean multipliers are updated using subgradient optimization algorithm.

Results show that both algorithms are quite acceptable and efficient. A comparison between the results of both algorithms shows considerable improvement compared to the result found in the literature. The enhancement of these results may be returned to the fact that step sizes are more suitably chosen and to the simplification done in the solution procedure.

Keywords: Energy, optimal scheduling, production planning, hybrid decomposition techniques.

INTRODUCTION

Energy is one of the most important products in life. Without energy, life would not become so easy and comfortable. Consumption of energy is found in different kind and forms. The production of fossil fuel energy (coal, oil and gas) is of major interest recent years. Statistics of 1999 show that 90.3% of the world energy consumption is from fossil fuel resources (11.1% of it is biomass energy). Electric, thermal and mechanical energy are typical products. Nuclear energy represents about 6.9% of the world energy consumption, while hydroelectric energy represents about 2.3%. On the other hand, renewable energy forms, which are important potential for the future, represent only about 0.5% of the world recent energy consumption) [1, 2].

Scheduling of energy production are decision processes which is concerned with the adoption and allocation of the natural and industrial resources in order to best satisfy marketing and customer requirements at minimum (production, setup, inventory, etc) costs. For a typical energy production and scheduling plan, the decision makers have to determine the total amount of energy to be produced (force level), scheduling of the overtime, the production run quantities and the sequencing of their occurrences.

Therefore, energy production planning and scheduling becomes a challenging problem for many reasons. The *first reason* is the structure of both the demand and costs which are usually variables (for real planning problems) over time according to seasonal structures. It is recommended in this scope to revise the energy demand (and accordingly the production) plans as soon as new information about these structures are available. The *second reason* is the difficulty to forecast precisely the energy demand at the very most detailed level. It is therefore seen to *aggregate* end or semi-end products (product items, trained personnel, supplies or accessories) which have similar production and marketing properties into *families*. For example, the family of crude oil energy may be considered as the aggregation of equipments, supplies, spare parts, replacements and accessories required for its production. Similarly, families for the natural gas energy, coal energy, hydroelectric energy, nuclear energy, biomass energy and renewable energy may be considered. Analogously, similar families are aggregated into *types* to simplify the planning problem. We can consider the aggregation of oil, natural gas and coal families into one type; the fossil fuel type. Also the hydro electric, solar, wind, geothermal, tidal, and biomass energy families can be aggregated into another type; the renewable energy type. A third type may contain families of trained labors, maintenance crews and management staff. Such an aggregation scheme is known as the hierarchical scheme [6].

The *third reason* is that there is less flexibility to modify the operating conditions of any energy production system. Such rigidity is met in the first line with labour management. Therefore production is to be planned within the available regular production time as long as it is possible and producing as less as possible in overtime. The *fourth reason* is that there are usually long and uncertain delays in obtaining the natural and industrial resources (new equipments, workers training, subcontracting capacities, raw materials‹ item components, etc.) as well as an incompressible procurement and manufacturing lead time.

It is therefore clear that the energy production plan can not be implemented efficiently with a short sight view. In this scope it is recommended to plan for production ahead of time and for as long time as possible. That is because long planning horizons give the suppliers the chance to plan efficiently and pass on their clients part of the cost reduced so obtained. For these purposes, the energy production planning and scheduling processes in a patch processing environment are concerned with the aquisition, utilization and setting of the available resources.

Engineers, researchers and management scientists have paid a great part of attention for developing energy production planning systems to support such decision making processes. Two different approaches for energy production planning and scheduling have been appeared in the literature. The first approach, called *the monolithic approach*, formulates the production planning problems as a large scale mixed-integer linear programming problems, see e.g. Dzielinski and Gomory [3], Lasdon and Terjung [4] and Manne [5]. The monolithic formulation of the planning problem is usually solved approximately each planning period with the instantaneous decision (of that period) being implemented. A common solution procedure for the monolithic approach is found to be equivalent to a Lagrangean relaxation to solve the dual problem to the mixed-integer linear program. The solution of the dual problem is then rounded to obtain a feasible near optimum solution to the main problem. The second approach, called *the hierarchical approach*, partitions the planning and scheduling problem into a hierarchy of smaller subproblems, see e.g. Hax and Meal [6]. The upper hierarchy deals with strategic decisions for the planning horizon, while the lower hierarchy deals with the more detailed short term scheduling. At any planning period, the subproblems are solved sequentially, with constraints imposed from the solution of the upper hierarchy on the lower hierarchy, and therefore, the approach implements also instantaneous solutions. However, there are theoretically no mathematical feedback processes to the upper hierarchy and therefore, exact optimality is not guaranteed.

In a short comparison between the two approaches, it is known that the monolithic approach focuses on a well defined model formulation for which the optimization process is meaningful. While, in contrast, the main problem is partitioned in the hierarchical approach into subproblems, each of which is separately solved, resulting in a system of suboptimization problems.

The hierarchical approach has four potential advantages over the monolithic approach. The *first advantage* is that in the monolithic formulation of the energy planning and scheduling problem, the solution is obtained (in most cases), as mentioned before, through linear programming algorithms applied to a large size mixed-integer programming problem. On the other hand, in the hierarchical approach, the solution is in general obtained by solving a dynamic programming problem only, see e.g. [7, 8], or additionally by solving another small size linear programming problem [9]. For this reason, the computational effort required for the monolithic approach is generally greater than that required for the hierarchical approach. The *second advantage* is that the hierarchical approach may require less detailed demand input data, in that it needs only aggregate demand data, over the planning horizon and detailed demand data, over a much shorter scheduling horizon. The monolithic approach requires usually detailed demand data over the whole planning horizon. The *third advantage* is the extent to which the hierarchical subproblems correspond to organizational and decision-making echelons. The consequences of this point are the increased interaction between the planning system and the decision makers at each level, and the improved coordination of the objectives throughout the organization. The *fourth advantage* is that uncertainty is better treated by this approach, because the more detailed scheduling decisions regarding a certain job are postponed until the instant before the job begins processing; see e.g. Aardal and Larsson [7], Graves [9] and Ari [10].

In this work, a comparison is presented between two different hybrid procedures for solving the hierarchical energy production planning and scheduling problem in the framework of Hax and Meal [6]. Furthermore, a simplification in one of them is made. The first procedure, presented by Graves [9], is a hybrid approach where hierarchical subproblems are included in an overall mixed-integer linear program formulation. Accordingly, the problem is partitioned into two hierarchies; in the lower hierarchy, energy items are aggregated into families, which, in turn, are aggregated into types in the upper hierarchy. In this procedure, the set of inventory constraints are priced out using Lagrangean multipliers in the main objective function, which is then partitioned into two subproblems. The first subproblem is an optimal control problem on the family level (lower hierarchy) while the other subproblem is a linear program on the type level (upper hierarchy). The two subproblems are linked together by an inventory aggregation constraint, i.e. the inventory on the type (aggregate) level is to be equal to the sum of the inventories on the family (detailed) level. By this approach, a feedback is included automatically in the solution procedure, but it does not pass directly from the lower level to the upper level since both subproblems are solved in parall.

In the second procedure, another hybrid approach is presented by Aardal and Larsson [7]. Instead of using Lagrangean relaxation with respect to a group of constraints in the original model, the model is first partitioned according to Benders decomposition [11]. The Benders cuts are then priced out by means of a set of Lagrangean multipliers into the subproblem generated on the family level. Accordingly, one subproblem is trivial and the other subproblem is an incapacitated lot-sizing optimal control problem. The size of this subproblem is exactly the same as that of the equivalent subproblem in the procedure of Graves. In both procedures, the optimal control subproblem is solved using dynamic programming [8] and the set of Lagrangean multipliers are updated using subgradient optimization algorithm.

MODEL FORMULATION

The main objective of the energy production planning and scheduling process is twofold. First *the planning function* of the system is to determine what resources are needed and at what points of time they are required, in order to satisfy the aggregate demand over a prespecified planning horizon. Second, *the scheduling function* should determine for the immediate scheduling period how the available resources should be allocated to the individual products in order to provide the best customer services at a minimum total cost. For these purposes, it is assumed that the demand is known over some horizon. Both the plan and the schedule should be revised periodically in a rolling-schedule fashion as soon as improved demand forecasts are obtained.

In this work, it is assumed that energy products could be grouped into two levels of aggregation. As discussed in the introduction, the energy end products to be delivered to the consumer are grouped into families. Items and accessories in one family share a common setup cost, and therefore need to be considered jointly when preparing a planning schedule. Families having the same demand pattern are aggregated into types (e.g. oil, gas and coal families build up the fossil fuel energy type). Families in one type (and consequently their items) share a common unit inventory holding cost. This aggregation scheme has been first proposed by Hax and Meal [6], and has been observed in many industrial situations. Interested readers may see [6, 10] for detailed discussion of this aggregation scheme of production planning.

In this section, we shall give a full discussion of the mathematical model of the hierarchical approach of production planning. The problem is to determine a production schedule on the family level, which in turn, provides the families inventories. Also, it is required to determine the production and inventory plans on the type level. These decisions should be made in order to minimize the total cost, i.e. the sum of the overtime cost, the inventory holding cost and the setup cost subject to production capacity and demand requirements constraints. It is also shown that both family and type decisions are included in the overall problem formulation, which is described below.

In the following formulation, subscript t corresponds to time periods, i to types, j to families and n denotes number of time periods of the planning horizon. The decision variables for this model are:

- O_t overtime required in period t,
- I_{it} (I_{jt}) inventory level of type i (family j) in period t,
- P_{it} (P_{jt}) production quantity of type i (family j) in period t,
- X_{jt} zero-one variable to indicate setup of family j in period t.

Input data to the model are:

- c_t production cost premium for overtime in period t,
- h_{it} inventory holding cost for type i in period t,
- s_{jt} setup cost for family j in period t,
- d_{it} (d_{jt}) demand for type i (family j) in period t,
- r_t regular production time available in period t,
- k_i production time for unit of type i,
- T_i set of families belonging to type i,
- m_{jt} maximum production quantity for family j in period t, i.e. $m_{jt} = \sum_{k\geq t} d_{jk}$

We consider the following simple model formulation:

(PPS)

$$\min Z = \sum_t \left(c_t O_t + \sum_i h_{it} I_{it} \right) + \sum_j \sum_t s_{jt} X_{jt} \tag{1}$$

subject to

$$P_{jt} + I_{jt} - I_{j,t+1} = d_{j,t+1} \qquad \forall j,t \tag{2}$$

$$\sum_i k_i P_{it} - O_t \leq r_t \qquad \forall t \tag{3}$$

$$\sum_{j\in T(i)} I_{jt} = I_{it} \qquad \forall i,t \tag{4}$$

$$\sum_{j\in T(i)} P_{jt} = P_{it} \qquad \forall i,t \tag{5}$$

$$P_{jt} - m_{jt} X_{jt} \leq 0 \qquad \forall j,t \tag{6}$$

$$O_t, P_{it}, I_{it}, P_{jt}, I_{jt} \geq 0 \qquad \forall i,j,t \tag{7}$$

$$X_{jt} = \{0,1\} \qquad \forall j,t \tag{8}$$

As shown in the aggregation scheme of the above model, production capacity costs (of the overtime) and inventory holding costs are accounted for by types, while production setup costs are accounted for by families. Constraint (2) is an inventory balance on the family level, while constraint (3) is a production capacity constraint. Constraints (4) and (5) are consistency constraints between types and families for inventory and production, respectively. They link the upper and lower hierarchies for both quantities. In the work of Graves [9], constraint (5) is replaced with an inventory balance on the type level, i.e.

$$P_{it} + I_{it} - I_{i,t+1} = d_{i,t+1} \qquad \forall i,t \tag{9}$$

which is already fulfilled since $d_{it} = \sum_{j\in T(i)} d_{jt}$ together with constraint (5). Constraint (6) relates the binary setup variables X_{jt} to the family production P_{jt}.

The model (PPS) is a single resource model for scheduling types and families and possibly the simplest of all planning models. It considers only one constraint production resource, and incorporates only a single option, the overtime, for varying the resource level. We ignore the scheduling of items within a family; this is partially justified by the aggregation scheme in that the total costs can be determined either at the type aggregate or at the family aggregate.

Many approaches of aggregation and solution procedures have been presented in the literature. We shall discuss two of these procedures in this paper and compare their differences and results. Readers who are interested in more details about such approaches and procedures may see for instance Aardal and Larsson [7] as well as Graves [9].

SOLUTION PROCEDURES

1. Using Lagrangean Multipliers

The procedure presented by Graves [9] examines Lagrangean relaxation to solve the dual problem to (*PPS*) by an iterative procedure. For this relaxation, Graves had priced out the set of inventory consistency constraints, constraint (4), into (*PPS*). Instead of constraint (4), we have priced out the set of production consistency constraints, constraint (5), into (*PPS*) in order to satisfy some conditions for the dynamic programming algorithm [12] used in our work. The dynamic programming algorithm requires concave cost functions (for production and inventory). Furthermore, lower and upper bounds are imposed on the set of Lagrangean multipliers in order to insure boundness of the objective function of the subproblem generated on the aggregate (types) level, as discussed later.

Having a set of Lagrangean multipliers $\lambda = \{\lambda_{it}\}$, a relaxation of (*PPS*) is obtained by pricing constraint (5), jointly with these multipliers into the objective function (1) as:

$$L(\lambda) = min \left[Z + \sum_{i,t} \lambda_{it} \left(\sum_{j \in T(j)} P_{jt} - P_{it} \right) \right] \quad (10)$$

subject to

$$\begin{cases} Z = \sum_t \left(c_t O_t + \sum_i h_{it} I_{it} \right) + \sum_j \sum_i s_{jt} X_{jt} \\ \text{and} \quad (2), (3), (4), (6), (7), (8). \end{cases} \quad (11)$$

The dual problem to (PPS) is found to be:

$$(D) \qquad max\ L(\lambda) \qquad (12)$$

The solution of the dual problem (D) is obtained by an iterative procedure working in the hierarchical framework of Hax and Meal [6]. This could be achieved by partitioning the relaxed function (10)–(12), as it is easily recognised, into two subproblems. The first subproblem is an aggregation model on the type level, and could be written as follows.

(*LP*)

$$min\ Z_{LP} = \sum_t \left[c_t O_t + \sum_i \left(h_{it} I_{it} - \lambda_{it} P_{it} \right) \right] \quad (13)$$

subject to (2), (3), (7)

The other subproblem of the Lagrangean relaxation (10)–(12) is found to be:

$$(DP) \quad min \quad Z_{DP} = \sum_{j,t} s_{jt} X_{jt} + \lambda_{it} P_{jt} \qquad (14)$$

subject to

$$\begin{cases} (4), (6), (7), (8) \\ \text{and} \quad i = T^{-1}(j) \end{cases} \quad (15)$$

where $i = T^{-1}(j)$ if and only if $j \in T(i)$. Clearly, (*DP*) is a disaggregation model dealing with scheduling of product families. Furthermore, (*DP*) could be separated by families into a set of incapacitated lot-sizing problems, each of which is easily solved by dynamic programming [8].

In order to insure boundness of the objective function of (*DP*) the Lagrangean multipliers λ_{it} should be positive. That is because negative multipliers, which are considered as cost coefficients in (*DP*), means to produce as much as possible to minimize its objective function, and consequently the value of the objective function will go to ∞. On the other side, λ_{it} must be less than or equal to $c_t k_i + \sum_{k \geq t} h_{ik}$ in order to keep boundness of the objective function of (*LP*). For these reasons, λ_{it} should fulfill the following inequality.

$$0 \leq \lambda_{it} \leq c_t k_i + \sum_{k \geq t} h_{ik} \qquad \forall i,t \qquad (16)$$

It is already known in such mixed-variables problems that a duality gap usually exists, i.e. the objective function of the dual problem, i.e. equation (12) above, is usually less than that of the relaxed problem, i.e. equation (10) above. That is because the relaxed problem does not have the integrality property, i.e. its optimal value will be changed by dropping the integrality properties on its variables. In this context, we have used the following definition of the duality gap *g*

$$g = \frac{\text{Best upper bound} - \text{Best lower bound}}{\text{Best upper bound}} \quad (17)$$

The solution procedure for this approach is organized as follows.

Algorithm A1

Step 0: Set $k = 0$ and initialize $\lambda_0 = \{\lambda_{it}\}$.
Stop 1: Solve the subproblem (*DP*) with $\lambda = \lambda_k$.
Step 2: Solve the subproblem (*LP*) with $\lambda = \lambda_k$.
Step 3: If some preset stopping criterion is met, *stop*, otherwise *goto step 4*.
Step 4: Update λ_k.
Step 5: Set $k = k + 1$ and *goto step 1*.

Initially, the λ_{it}'s could be set to zero. As mentioned before, the problem (*DP*) could be solved for each family using dynamic programming, while (*LP*) is solved using the simplex method. The sum of the objective functions of both (*DP*) and (*LP*) from *steps 1* and *2*, constitutes a lower bound to (*PPS*). An upper bound of (*PPS*) is obtained by aggregating the production schedule obtained from *step 1* and then substituting this aggregate production plan into the objective function (1) of (*PPS*). The procedure can be terminated once both solutions from steps 1 and 2 become identically. Also the procedure may be terminated when a preset value of the duality gap is reached, i.e. both the lower and upper bounds become close enough to each other, or after doing a fixed number of iterations. Updating λ_k in *step 4* could be obtained using subgradient optimization algorithm, which will be discussed briefly later.

2. Using Benders Decomposition

In this section we present the solution procedure used to solve (*PPS*) in the framework of Aardal and Larsson [7]. They had partitioned the problem according to Benders decomposition [11], with some of the variables are classified as complicating variables, namely X_{jt}, I_{jt}, and P_{jt} in (*PPS*) above. Accordingly, (*PPS*) is partitioned into a *subproblem* on the type level and a *master problem* on the family level. The subproblem is to minimize the sum of the production (overtime) cost and the inventory holding cost at each time period, i.e.

$$(SP) \quad Z_{SP}(t) = \min \; c_t O_t + \sum_i h_{it} I_{it} \quad (18)$$

$$\text{subject to} \qquad (2)-(5), (7)$$

The subproblem (*SP*) is trivial and has the following feasible and unique primal solution which could be easily seen by inspection.

$$\begin{cases} P_{it} = \sum_{j \in t(i)} P_{jt} & \forall i,t \\ I_{it} = \sum_{j \in t(i)} I_{jt} & \forall i,t \\ O_t = \max\left\{0, \sum_i k_i \sum_{j \in t(i)} P_{jt} - r\right\} & \forall i \end{cases} \quad (19)$$

Introducing the dual solution to (*SP*), the following formulation of the master problem is obtained.

$$(MP) \quad Z_{MP} = \min \sum_{j,t} s_{jt} X_{jt} + \sum_{j,t} h_{it} \sum_{j \in T(i)} I_{jt} + \sum_t q_t \quad (20)$$

subject to

$$q_t \geq \sum_i u_{it} \sum_{j \in T(i)} P_{jt} + v_t r_t \qquad \forall t, \quad (21)$$

$$q_t \geq 0 \qquad \forall t \quad (22)$$

$$\text{and } (2), (6), (7), (8)$$

where, q_t is an auxiliary variable of the Benders master problem, u_{it} and v_t are dual variables of the subproblem (*SP*) whose values are:

$$u_{it} = -k_i v_t, \qquad v_t = \begin{cases} -c_t & \text{if } O_t > 0 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

respectively.

Now, a vector of multipliers $\lambda = \{\lambda_t\}$ will be introduced in order to price out the first constraint of the master problem (*MP*) into its objective function. The authors [7] had also imposed the following lower and upper bounds on these multipliers in order to simplify the relaxed model of (*MP*).

$$0 \leq \lambda_t \leq 1 \qquad \forall t \quad (24)$$

Finally, [7] arrived at the following formulation of the relaxed master problem.

$$(RMP) \quad Z_{RMP}(\lambda) = \min \sum_{j,t} s_{jt} X_{jt} + \sum_{i,t} h_{it} \sum_{j \in T(i)} I_{jt} + \sum_t \lambda_t \left(\sum_i u_{it} \sum_{j \in T(i)} P_{jt} + v_t r_t \right) \quad (25)$$

$$\text{subject to } (2), (6), (7), (8)$$

According to the theory of Lagrangean duality for integer programming, Aardal and

Larsson [7] had obtained the greatest lower bound for (*PPS*) by solving the following dual problem to the relaxed master problem.

$$(DRMP)\ Z_{DRMP} = \max\ Z_{RMP}(\lambda) \quad \text{subject to (16).} \qquad (26)$$

In this paper, the model of Aardal and Larsson [7] is simplified in the following way. As it could be easily seen, the subproblem (*SP*) as well as its solution is trivial. Instead of using this decomposition technique, we substitute the sets of constraints (4) and (5) directly into the objective function of (*PPS*). Using the physical definition of O_t. as shown in (19) and defining

$$a_{it} = k_i y_t, \qquad y_t = \begin{cases} c_t & \text{if } O_t > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (27)$$

then, we get the production cost (of the overtime) $b_t := c_t O_t$ as:

$$b_t = \sum_i a_{it} \sum_{j \in T(i)} P_{jt} - y_t r_t \qquad \forall t \qquad (28)$$

Now, we price out this constraint, which replaces constraint (3), jointly with the previously defined vector of multipliers λ into the objective function of (*PPS*) after eliminating P_{it} and I_{it}. We obtain the following relaxed model.

$$\min \sum_t \left[\left(b_t + \sum_i h_{it} \sum_{j \in T(i)} I_{jt} + \sum_j s_{jt} X_{jt} \right) + \lambda_t \left(\sum_i a_{it} \sum_{j \in T(i)} P_{jt} - y_t r_t - b_t \right) \right] \qquad (29)$$

subject to (2), (6) – (8).

Again by imposing lower and upper bounds on λ as shown in (16), we get the following relaxed problem for (PPS).

$$(RPPS)\quad Z_{RPPS}(\lambda) = \min \sum_{j,t} s_{jt} X_{jt} + \sum_{i,t} h_{it} \sum_{j \in T(i)} I_{jt} + \sum_t \lambda_t \left(\sum_i a_{it} \sum_{j \in T(i)} P_{jt} - y_t r_t \right) \qquad (30)$$

subject to (2), (6) – (8)

which is exactly the same relaxed model (*RMP*), Eq. (25). The highest lower bound is again obtained by maximizing the dual problem to (*RPPS*), which is the same as (*DRMP*), while the upper bound is obtained directly from the objective function (1) of (*PPS*). Although $Z_{DRMP} < Z_{RPPS}$, due to the existence of the duality gap, (*DRMP*) has shown to give a lower bound on (*PPS*) which is much more tight to the optimal solution than a linear program formulation of the problem. The solution procedure of this approach is as follows.

Algorithm A2

Step 0: Set $k = 0$ and choose an initial value for $\lambda_0 = \{\lambda_k\}$.

Step 1: Solve the relaxed problem (*RPPS*) with $\lambda = \lambda_k$.

Step 2: Update λ_k.

Step 3: If some preset stopping criterion is met, then *stop*, otherwise, set $k = k + 1$ and *goto step 1*.

An acceptable initial value is to set $\lambda_0 = 0$. The relaxed problem (*RPPS*) is an optimal control problem and separates by families, for a certain value of λ, into a set of incapacitated lot-sizing problems, each of which is easily solved by dynamic programming. The procedure may terminate after doing a fixed number of iterations or if a preset duality gap is reached, i.e. both the lower and upper bounds obtained from *steps 1* and *2* are close enough to each other. Another termination criterion is when the subgradient of λ becomes zero. As mentioned before, the updating of λ is, at best, done using subgradient optimization. It is also seen that a lower bound to (*PPS*) could be obtained from *step 1* in Algorithm A2.

3. Subgradient Optimization

A standard method for solving the dual problems arising in the above procedures, problem (*D*) in Eq. (11) and problem (*DRMP*) in Eq. (26), is the subgradient optimization algorithm [13], in which a dual solutions are updated according to

$$\lambda_{k+1} = \lambda_k + \theta_k \gamma_k \qquad (310)$$

where, for a certain value of λ, γ represents the subgradient of the relaxed problem at hand, θ is a step size and k is the iteration counter.

From the Lagrangean multipliers procedure of Graves [9] presented above, the relaxed problem has a subgradient $\gamma = \{\gamma_{it}\}$ which could be expressed as

$$\gamma_{it} = \sum_{j \in T(i)} P_{jt} - P_{it} \qquad \forall i,t \qquad (32)$$

The respective step size is calculated as

$$\theta = \frac{\rho[w - Z(\lambda)]}{\|\gamma\|^2} \qquad (33)$$

where, w represents the current best upper bound obtained, $Z(\lambda)$ is the current lower bound and ρ is a step size parameter.

In the Benders decomposition approach presented above, the relaxed problem has a subgradient $\gamma = \{\gamma_t\}$ which could be found as follows. Let $a_t = \{a_{it}\}$ and $P_t = \{P_{it}\}$, then

$$\gamma_t = \begin{pmatrix} a_t \\ y_t \end{pmatrix}^T \begin{pmatrix} P_t \\ -r_t \end{pmatrix} \qquad (34)$$

The choice of the step size parameter ρ in (33) is critical to the behavior of the solution procedure used. For convergence to occur and to avoid cycling near the solution, the step size is chosen such that the step size tends to zero and the sum of the sequence of step sizes goes to infinity. For the solution procedure of Algorithm A1, we had obtained the best results by using the following formula

$$\rho = \frac{1}{ln\left(\sqrt[4]{k} + 2\right)} \qquad (35)$$

where k, as before, is the iteration counter. On the other hand, we had found that the following formula is more suitable to the solution procedure of Algorithm A2.

$$\rho = \frac{1}{\left(max\{40, k\} - 39\right)^{0.5 + \frac{k}{2k+200}}} \qquad (36)$$

For other step size strategies, see e.g. [13, 14].

COMPUTATIONAL STUDY AND IMPLEMENTATIONS

Case Studies

In this work, four sets of problems are used as case studies for testing the procedures discussed above. Each problem set consists of nine problems. Each problem in *the first two sets* of problems constitutes twenty families, which, in turn, are aggregated into three types with five families in each of the first two types and ten families in the third type. *The third and fourth sets* of problems contain product structure of forty families aggregated also into three types with ten families in each of the first two types and the remaining twenty families in the third type. The number of time periods n is twelve for all test problems. The overtime costs and the inventory holding costs for the three types are given the following values for all time periods.

$$\left.\begin{array}{l} c_t = 5.0 \\ h_{1t} = 1.0, \quad h_{2t} = 1.75 \quad \text{and} \quad h_{3t} = 1.5 \end{array}\right\} \forall t.$$

The setup cost as well as the resource coverage, i.e. the available regular time, is assumed varied for all test problems. The setup cost may have three different levels with the lower level having one fifth the value of the medium level and the higher level is five times the medium level. The value of the setup cost at the medium level at each time period for each family belonging to type i is generated randomly from the uniform distribution over the range $\left(\underline{S}_i, \overline{S}_i\right)$ given in Table 1 below.

The levels of available regular time cover 80%, 100% and 120% of the time required for the total demand over the planning horizon which could be expressed as

$$r_t = \tfrac{1}{n} \sum_{i,t} k_i d_{it}$$

and is constant for all time periods, where, k_i, the unit production time for type i are given the following values:

$k_1 = 1.0, \quad k_2 = 2.0$ and $k_3 = 1.5$

The demand for family j in time period t is given by

$d_{jt} = f_{it}\, u_j$ and $j \in T(i)$

Table 1 Setup costs Ranges of medium level.

Type	1	2	3
$\underline{S}_i$	50	100	100
$\overline{S}_i$	150	200	200

where f_{it} is the seasonality demand factor for families belonging to type i in period t and u_j is the normal demand of family j. No seasonal variation in demand is considered for problems in the first and third sets of problems (i.e. f_{it} = 1.0 for all types in all time periods) while seasonality factors for the second and fourth sets of problems take the values in Table 2.

The normal demand u_j of family j is drawn randomly from the uniform distribution over the ranges $(\underline{D}_j, \overline{D}_j)$. The ranges for families 1 – 20 are given in Table 3, and these ranges are the same for families 21 – 40.

Finally, the initial inventory I_{j0} of family j is drawn randomly from the uniform distribution over the range (0, Q_j) where Q_j represents the economic order quantity of the family, and is based on its average demand rate, holding cost and the setup cost.

Table 2 Demand seasonality factor for second and fourth sets of problems.

Time period	1	2	3	4	5	6	7	8	9	10	11	12
Type 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Type 2	0.8	0.8	0.7	0.5	0.7	1.0	1.0	1.2	1.3	1.5	1.2	1.0
Type 3	1.0	1.0	1.0	1.2	1.3	1.5	1.3	1.0	0.9	0.7	0.6	0.8

Table 3 Ranges for the normal demand u_j for the first 20 families.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\underline{D}_j$	20	40	15	25	80	80	15	40	40	55	20	20	30	30	20	30	30	50	60	40
$\overline{D}_j$	40	60	25	65	120	120	25	60	60	85	40	40	50	50	40	70	50	100	90	80

Implementations

An efficient optimal control algorithm for production planning due to Proth [12] (chap. 1) is used to solve the problems (*DP*) and (*RPPS*) of this study. The algorithm is much faster than the usual dynamic programming procedures of Wagner and Whitin [8]. The algorithm gives an optimal production plan for each family in each call, i.e. it is called number of times equal to number of families each iteration. At each iteration, the solution to the dual problem obtained from the subgradient optimization algorithm is projected according the respective projection condition, i.e. conditions (16) and (24); respectively, to enforce the values of the Lagrangean multipliers to lie within the bounds mentioned above.

The revised simplex method is used to solve the linear programming problem of the Lagrangean multipliers approach. For this method, we feed the simplex algorithm in the first iteration with a basic feasible solution. This initial feasible solution is set so as to produce at each time period just to cover the required demand in that period if the inventory level is not enough. Then, the feasible solution obtained from the output of the simplex algorithm after each iteration is used as an input feasible solution for the next iteration. Similarly, the inverse of the input basic matrix required for the simplex algorithm is initialized making use of its special block structure, and its update obtained from the output of the algorithm is again used as input to the next iteration. In order to reduce the effect of error propagation on the results, the input feasible solution as well as the input inverse of the basic matrix are initialized each number of iterations, which is selected as twenty iterations in our implementations. All implementations are done using codes made by the author.

RESULTS AND DISCUSSIONS

The duality gap parameter (17), which is used for convergence tests in our implementations, represents the maximum deviation from the optimum solution expected. Tables 4 – 7 report the percentage duality gap (i.e. 100 g) obtained from the solution procedure described in Algorithm A1, while Tables 8 – 11 report those results obtained from the procedure described in Algorithm A2. For each one of the thirty six problems studied, we have made ten runs for each problem with the same resource coverage and setup cost factors, but with different random generations of the data, and the figures represented in the tables below are the mean values of these ten of runs.

It has been observed that both algorithms had reached a preset value of the duality gap (less than or equal to 0.1% in our case) in some problems (specially in those cases where high resource coverage is chosen with low or medium setup costs), but after much fewer iterations with Algorithm A2 than Algorithm A1. In this concern, it has been observed that one iteration of Algorithm A1 requires computational time (CPU time) which ranges from six to fifteen times the time required for one iteration of Algorithm A2, depending on the operational mode of the computer system used and on the type of problem (problem sets 1 and 2 have higher ratios than sets 3 and 4). Also in Algorithm A2, out of 360 test examples, the exact optimal solution (i.e. zero duality gap) is reached in 29 of them, while in another 41 test examples, a duality gap less than 0.1% is obtained. On the other hand, using Algorithm A1, a duality gap less than 0.1% is obtained in 25 test examples and no exact solution is obtained. In general, the results of all of the thirty six problems but nine obtained from Algorithm A1 are better than those obtained from Algorithm A2.

A comparison between the results of Algorithm AI and those presented by Graves [9] is presented in Table 12. The deviation of all test problems but six have better enhancements against those presented in [9]. On the other hand, a comparison between the results obtained from Algorithm A2 and those of Aardal and Larsson [7] is presented in Table 13. Also the deviation of all test problems but six have better enhancements against those of [7]. The enhancement of these results against those of Aardal and Larsson [7] and Graves [9] may be returned to the fact that step sizes are more suitably chosen, to the simplification done in the solution procedure presented in this study and also to the fact that we have done relatively more iterations than in [7] and [9].

Results of Algorithm A1

Table 4 Percentage duality gap for problem set 1 (constant demand).

Setup cost factor	Resource coverage, %		
	80	100	120
0.2	0.798	2.229	0.264
1.0	0.689	3.190	0.103
5.0	1.005	2.485	1.073
Average	0.831	2.635	0.480

Table 5 Percentage duality gap for problem set 2 (seasonal demand).

Setup cost factor	Resource coverage,%		
	80	100	120
0.2	1.056	3.256	0.293
1.0	1.188	3.100	0.171
5.0	0.749	2.232	1.020
Average	0998	2.863	0.495

Table 6 Percentage duality gap for problem set 3 (constant demand).

Setup cost factor	Resource coverage, %		
	80	100	120
0.2	0.620	1.027	0.164
1.0	0.513	1.135	0.112
5.0	0.312	1.199	0.326
Average	0.482	1.120	0.201

Table 7 Percentage duality gap for problem set 4 (seasonal demand).

Setup cost factor	Resource coverage,%		
	80	100	120
0.2	1.079	3.269	0.186
1.0	0.918	1.979	0.133
5.0	0.307	0.980	0.453
Average	0.768	3.114	0.257

Results of Algorithm A2

Table 8 Percentage duality gap for problem set 1 (constant demand).

Setup cost factor	Resource coverage, %		
	80	100	120
0.2	1.733	2.489	0.000
1.0	0.805	3.541	0.091
5.0	1.839	2.757	1.308
Average	1.459	2.929	0.466

Table 9 Percentage duality gap for problem set 2 (seasonal demand).

Setup cost factor	Resource coverage,%		
	80	100	120
0.2	2.707	3.936	0.077
1.0	1.263	3.467	0.572
5.0	1.421	2.873	1.326
Average			

Table 10 Percentage duality gap for problem set 3 (seasonal demand).

Setup cost factor	Resource coverage, %		
	80	100	120
0.2	0.703	1.385	0.000
1.0	0.324	1.208	0.038
5.0	0.524	1.352	0.543
Average	0.517	1.315	0.194

Table 11 Percentage duality gap for problem set 4 (seasonal demand).

Setup cost	Resource coverage,%		
	80	100	120
0.2	1.005	2.460	0.100
1.0	0.422	2.249	0.182
5.0	0.427	1.238	0.819
Average	0.618	1.982	0.367

Table 12 Comparison with the work of Craves [9].

	Deviation, %			Problems with		Averages of problems, %			
	Min.	Max.	Mean	g>2.5%	g>2.0%	Set 1	Set 2	Set 3	Set 4
Present work	0.0	3.269	1.101	4	7	1.315	1.452	0.601	1.034
Graves	0.2	4.4	2.2	14	19	1.9	2.0	2.8	1.8

Table 13 Comparison with the work of Aardal and Larsson [7].

	Deviation, %			Problems with		Averages of problems, %			
	Min.	Max.	Mean	g >2.5%	g>2.0%	Set 1	Set 2	Set 3	Set 4
Present work	0.0	3.936	1.311	6	9	1.618	1.960	0.675	0.989
Aardal, et al	0.0	5.95	2.34	13	18	2.27	2.54	1.93	2.60

REFERENCES

1. *World Energy Consumption, International Energy Outlook 2001*, EIA, Web site http://www.eia.org, Jan. 2001.
2. Khalil, A., *Introduction to Environmental Studies*, Tanta University, Tanta, Egypt, 2002.
3. Dzielinski, D.P., and Gomory, R.E., *Optimal Programming or Lot Sizes, Inventory and Labor Allocation*, Management Science 11(1965), pp. 874–890.
4. Lasdon, L.S., and Terjung, R.C., *An Efficient Algorithm For Multi-item Scheduling*, Operations Research, Vol. 19, No. 4 (July-Aug. 1971), pp. 946–969.
5. Manne, A.S., *Programming of Economic Lot Sizes*, Management Science, Vol. 4, No. 2 (Jan.1958), pp. 115–135.
6. Hax, A.C., and Meal, H.C., *Hierarchical Integration of Production Planning and Scheduling*, in North-Holland/TIMES, Studies in Management Science. Vol. 1, Logistics, North- Holland, Amsterdam and American Elsevier, New York, 1975, pp. 53–69.
7. Aardal, K., and Larsson, T,, *A Benders Decomposition Based Heuristic For The Hierarchical Production Planning Problem*, European Journal of Operational Research 45 (1990), pp. 4–14.
8. Wagner, H.M., and Whitin, T.M., *Dynamic Version of the Economic Lot Size Method*, Management Science, Vol. 5(1958), pp. 89–96.
9. Graves, S.C., *Using Lagrangean Techniques to Solve Hierarchical Production Planning Problems*, Management Science, Vol. 28, No. 3 (March 1982), pp. 260–275.
10. Ari, E.A., *Multilevel Approaches to Production Planning and Inventory Control*, Liecetiate Thesis, 1986; 017L, Lulea University of Technology, 1986.
11. Benders, J.F., *Partitioning Procedures For Solving Mixed-Variable Programming Problems*, Numerische Mathematik 4 (1962), pp. 238–252.
12. Bensoussan, A., Crouhy, M., and Proth, J.M., *Mathematical Theory of Production Planning*, Advanced Series in Management, North-Holland, Amsterdam, 1991.
13. Held, M., Wolfe, P., and Crowder, H.P., *Validation of Subgradient Optimization*, Mathematical Programming 6 (1974), pp. 62–88.
14. Coffin, J.L., *On Convergence of Subgradient Methods*. Mathematical Programming, 13(1977), pp. 329–347.