Planning Production Line Capacity to Handle Uncertain Demands for a Class of Manufacturing Systems with Multiple Products

Hao Liu, Qianchuan Zhao, *Senior Member, IEEE*, Ningjian Huang, Xiang Zhao

Abstract **Manufacturing systems flexibility is critical to meet the challenges arising from the uncertain demands. In this paper, a framework is proposed to evaluate the effect of utilizing flexibility in manufacturing systems which have the capability of producing multiple products at multiple plants. After transformation, a nonlinear polynomial programming model was proposed to describe the production planning problem, and a Reformulation-Linearization/convexification Technique (RLT) based branch-and-bound algorithm[1, 2] is applied to globally optimize the problem. A numerical example is provided to demonstrate the algorithm. At the end, we give the formulation of a stochastic programming model concerning uncertain demands.**

Keywords: Distributed Manufacturing Approaches and Practices; Flexible Manufacturing Systems; Enterprise Resource Planning Systems.

I. Introduction

MASS customization has been recognized as one of the new paradigms for today's manufacturing. As many companies increase their manufacturing flexibility, a plant possesses capability of producing multiple products in order to meet rapidly changing markets and gain competitive advantages. As a result of such a paradigm change, companies having a global manufacturing network need urgently a planning tool that helps to allocate production lines into multiple plants and to find an optimal solution of line configuration concerning the market demands.

Motivated by practical backgrounds, this paper is exploring ways to meet these challenges by introducing a framework that could utilize flexibility in manufacturing systems. The paper is organized as follows. In section II, we describe the planning problem of a manufacturing network in detail and proposed a general model for that. In section III, we introduce a RLT based branch-and-bound algorithm, and applied the algorithm to our model. A numerical example is given to demonstrate the application of the algorithm. In section IV, based on the model proposed in section II, we give the formulation of a stochastic programming model concerning uncertain demands. The paper is concluded in section V.

The work was supported by a contract between General Motors Company and Tsinghua University. Qianchuan Zhao received additional support from NSFC Grant Nos. (60574067, 60736027, 60721003)

Hao Liu and Qianchuan Zhao are with the Center for Intelligent and Networked Systems (CFINS), Department of Automation, TNList Lab, Tsinghua University, Beijing 100084, China

Ningjian Huang and Xiang Zhao are with Global Research & Development, General Motors Company, Warren, MI 48090, USA

Corresponding author: Qianchuan Zhao (zhaoqc@mails.tsinghua.edu.cn)

II. The Planning Problem: A General Model

In this section, we propose a general model of the planning problem of a manufacturing network. Inspired by the model of BMW's Global Production Network reported in [3], we consider a flexible manufacturing network in which several plants are built up with given capability of producing certain but not necessarily all kinds of the demanded products (see Figure 1), where the capability of a plant is represented by the maximal feasible total working hours of the plant, that is, normal working hours plus overtime hours. Different plants may have different maximal production line rates for different products, which are also given, and the producing processes of different products in a certain plant share the production capacity of the plant. Therefore, the planning problem is defined as when and where, i.e., in which plant(s), to produce what product(s) with what production line rates in order to meet the demands best and maximize the total profit under the constraints of capacity limit during all the planning periods. Here periods may be years, months or weeks, etc., and in our work we consider periods as weeks.

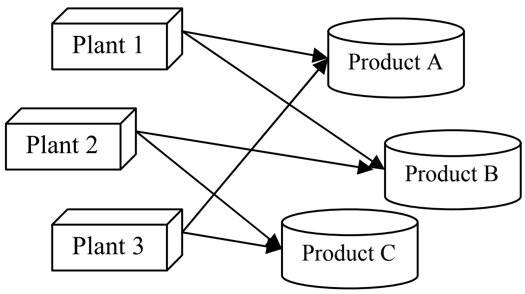

Figure 1: An example of manufacturing network. In this network, each of the three plants can respectively produce two of the three products.

For a plant that is capable of producing multiple products, its actual production line rate would vary either when we slow down or speed up production or when we stop producing one product and switch the production to another product with a different line rate, on the condition that the actual production line rate will not exceed the maximal rate. When the production line rate changes, some required costs such as costs for facilities uploading, tooling exchange, line maintaining, etc., which we call set-up cost, will be incurred. Corresponding with reality, we would use a piece linear function to describe the relationship between the set-up costs and actual production line rate (see figure 2).

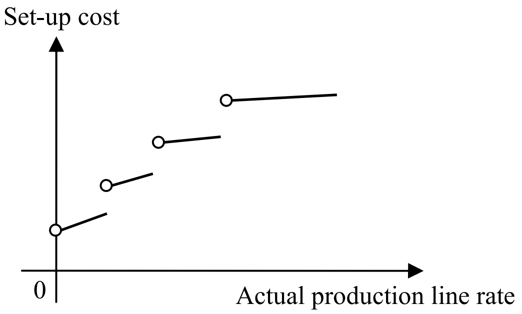

Figure 2: The relationship of set-up cost and actual production line rate.

Besides, we assume in our model that there is no inventory, and underproduction will cause a penalty. We consider more seriously the underproduction happening in the near term than in the future, so a discount factor of penalty is introduced. The formulation for the planning problem is proposed as follows.

Given Values:

$i \in I$ index of a plant. $I = \{1, 2, \ldots, \text{number of plants}\}$.
$j \in J$ index of a product. $J = \{1, 2, \ldots, \text{number of products}\}$.
$t \in T$ index of a period. $T = \{1, 2, \ldots, \text{number of periods}\}$.
JPH_{ij0} maximal line production rate (or Job Per Hour) of the production line of product j in plant i. $JPH_{ij0} = 0$ if there is no production line of product j in plant i.
HIW_{it}^{n} (HIW_{it}^{o}, respectively) maximal normal working hours (overtime hours, respectively) of plant i in period t. Here HIW stands for Hours In a Week, since we consider the periods as weeks in our work. Values of HIW_{it}^{n} and HIW_{it}^{o} depend on the shift pattern adopted: HIW_{it}^{n} = (Hours/shift) × (Shifts/day) × (Normal Days/week); HIW_{it}^{o} = (Hours/shift) × (Shifts/day) × (Overtime Days / week), in plant i in period t. see Table 1. HIW_{it}^{n} and HIW_{it}^{o} represent the production capacity of plant i in period t.

Table 1: Values of HIW corresponding to some commonly used shift patterns

Shift pattern No.	Shifts /day	Normal			Overtime		
		Hrs /shift	Days /week	**HIW^n (Hrs)**	Hrs /shift	Days /week	**HIW^o (Hrs)**
1	3	8	5	**120**	0	0	**0**
2	2	10	6	**120**	0	1	**0**
3	3	8	5	**120**	8	1	**24**
4	2	10	6	**120**	1	6	**12**

sc_{ijt} set-up cost for starting the production line of product j at plant i in period t, which could be a piecewise linear function of JPH_{ijt} (to be introduced later). For a given period t, sc_{ijt} equals to 0 if the product j is not being produced at plant i in the period t.
cc_{ijt} consumption cost per product j produced in plant i in period t, including cost of material and energy consumption, facility wearing and tearing, etc.
lc_{ijt}^{n} (lc_{ijt}^{o}, respectively) labor cost per product j produced in plant i in period t in normal working hours (overtime hours, respectively). lc_{ijt}^{o} is larger than lc_{ijt}^{n} due to the extra payment for workers (see Figure 3).

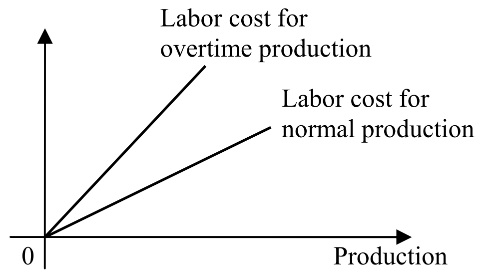

Figure 3: The relationship of labor cost and production.

p_j coefficient of penalty for underproduction of product j.
α discount factor of penalty for underproduction.
w_{jt} selling price per product j in period t.
d_{jt} demand for product j in period t.

Decision Variables:

JPH_{ijt} actual production line rate of the production line of product j in plant i in period t.
x_{ijt}^{n} (x_{ijt}^{o}, respectively) production of product j in plant i in period t in normal working hours (overtime hours, respectively).

Before giving the objective function and constraints, we first write the following expressions for clarity:
Total production cost:

$$C = \sum_{i\in I, j\in J, t\in T} \left[\left(cc_{ijt} + lc_{ijt}^{n}\right) x_{ijt}^{n} + \left(cc_{ijt} + lc_{ijt}^{o}\right) x_{ijt}^{o} + sc_{ijt} \right]$$

Total penalty of underproduction:

$$P = \sum_{j\in J} \left[p_j \sum_{t\in T} \alpha^{t-1} \left(d_{jt} - \sum_{i\in I} \left(x_{ijt}^{o} + x_{ijt}^{n} \right) \right) \right]$$

Total revenue:

$$W = \sum_{i\in I, j\in J, t\in T} (x_{ijt}^{n} + x_{ijt}^{o}) w_{jt}$$

We consider a total expense as

$$E = C + P$$

and the profit is $(W - E)$. Combining the similar terms, we have

$$W - E = \sum_{i\in I, j\in J, t\in T} \left(r_{ijt}^{n} x_{ijt}^{n} + r_{ijt}^{o} x_{ijt}^{o} \right) - \sum_{i\in I, j\in J, t\in T} sc_{ijt} - \sum_{j\in J, t\in T} p_j \alpha^{t-1} d_{jt}$$

where,

$$r_{ijt}^{n} = p_j \alpha^{t-1} + w_{jt} - cc_{ijt} - lc_{ijt}^{n},$$
$$r_{ijt}^{o} = p_j \alpha^{t-1} + w_{jt} - cc_{ijt} - lc_{ijt}^{o},$$

and the term $-\sum_{j\in J,t\in T} p_j \alpha^{t-1} d_{jt}$ has no effect on the decision (it is the same with the uncertain demands in section IV.) and then can be omitted. To maximize the profit, we obtain a programming model as (P):

$$\max\left[\sum_{i\in I,j\in J,t\in T}\left(r_{ijt}^n x_{ijt}^n + r_{ijt}^o x_{ijt}^o\right) - \sum_{i\in I,j\in J,t\in T} sc_{ijt}\right]$$

subject to

$$\begin{cases} JPH_{ijt} \le JPH_{ij0} & \forall i\in I, j\in J, t\in T \quad (1)\\ x_{ijt}^n \le HIW_{it}^n JPH_{ijt} & \forall i\in I, j\in J, t\in T \quad (2)\\ x_{ijt}^o \le HIW_{it}^o JPH_{ijt} & \forall i\in I, j\in J, t\in T \quad (3)\\ \sum_{j\in J}\dfrac{x_{ijt}^n}{JPH_{ijt}} \le HIW_{it}^n & \forall i\in I, t\in T, JPH_{ijt}\ne 0 \quad (4)\\ \sum_{j\in J}\dfrac{x_{ijt}^o}{JPH_{ijt}} \le HIW_{it}^o & \forall i\in I, t\in T, JPH_{ijt}\ne 0 \quad (5)\\ \sum_{i\in I}\left(x_{ijt}^n + x_{ijt}^o\right) \le d_{jt} & \forall j\in J, t\in T \quad (6)\\ JPH_{ijt}, x_{ijt}^n, x_{ijt}^o \ge 0, & \forall i\in I, j\in J, t\in T \quad (7) \end{cases}$$

where, sc_{ijt} in the objective function is a step function of JPH_{ijt}; constraint (1) means the actual line production rate will not exceed the maximal; constraint (2) ((3), respectively) leads to that the corresponding production within normal working (overtime, respectively) hours reduces to zero if JPH_{ijt} = 0; constraint (4) ((5), respectively) means the total normal working (overtime, respectively) hours of producing all the possible products in a plant cannot surpass the capacity of the plant; and constraint (6) means no overproduction is permitted. This programming problem is nonlinear and hard to solve due to the fractional terms in the constraints. We will introduce in the next section an algorithm to obtain the global optimal solution for the problem.

III. SOLVING THE PROBLEM: A RLT BASED BRANCH-AND-BOUND ALGORITHM

In this section we first introduce in detail a Reformulation-Linearization/convexification Technique (RLT) based branch-and-bound algorithm to solve the programming problem proposed in section II, and then apply the algorithm to problem (P). A simple numerical example is given to demonstrate the process of the algorithm.

A. Introduction to the RLT based branch-and-bound algorithm

The RLT based branch-and-bound algorithm was developed basically to solve polynomial programming problems having nonconvex polynomial objective and constraint functions. We here give an introduction of the algorithm following its developers Sherali and Tuncbilek. For more details of the algorithm, please see [1, 2]. Consider a polynomial programming problem given as:

$$\text{PP}(\Omega)\text{: } \min\left\{\phi_0(x) : x\in Z\cap\Omega\right\}$$

where,

$$Z=\left\{x:\phi_r(x)\ge\beta_r \text{ for } r=1,\cdots,R\right\},$$

$$\Omega=\left\{x: 0\le l_j\le x_j\le u_j<\infty \text{ for } j=1,\cdots,n\right\},$$

and

$$\phi_r(x)=\sum_{t\in T_r}\alpha_{rt}\left(\prod_{j\in J_{rt}} x_j\right) \text{ for } r=0,1,\cdots,R.$$

T_r is an index set of the terms in ϕ_r and α_{rt} is real coefficients of the polynomial terms $\prod_{j\in J_{rt}} x_j$ for $t\in T_r$, $r=0,1,\cdots,R$. J_{rt} is the indices set of corresponding polynomial term. Let $N=\{1,\cdots,n\}$, δ = the maximum degree of any polynomial term appearing in PP(Ω), and $\bar{N}=\{N,\cdots,N\}$ is composed of δ replicates of *N*. Then $J_{rt}\subseteq\bar{N}$, with $1\le|J_{rt}|\le\delta$, for $t\in T_r$, $r=0,1,\cdots,R$.

RLT Procedure (including two phases) [1, 2]:

1. Reformulation Phase. In this phase additional implied constraints are appended to the problem. For each $j\in N$, *bounding factors* are defined as $(x_j-l_j)\ge 0$ and $(u_j-x_j)\ge 0$, and the implied *bound-factor products* are:
$$F_\delta(J_1,J_2)=\prod_{j\in J_1}\left(x_j-l_j\right)\prod_{j\in J_2}\left(u_j-x_j\right)\ge 0 \quad (8)$$
where, $J_1\cup J_2\subseteq\bar{N}$ and $|J_1\cup J_2|=\delta$. Reformulation is done by adding constraints of type (8) to PP(Ω). In fact, we only need to add to PP(Ω) those bound-factor products constraints that involve polynomial terms that exist in PP(Ω).
2. Linearization/convexification Phase. In this phase new variables are introduced to replace the variable-product terms in the reformulated problem, that is, substitute
$$X_J=\prod_{j\in J} x_j \quad \forall J\subseteq\bar{N} \quad (9)$$
where the indices in *J* are sequenced in nondecreasing order, $X_{\{j\}}=x_j\ \forall j\in N$, and $X_\varnothing=1$.

The RLT procedure results in a *linear programming relaxation problem* LP(Ω). It is reported in [1, 2] that solving LP(Ω) yields a lower bound, or possibly a feasible or even optimal solution of PP(Ω). Based on this characteristic, a branch-and-bound algorithm can be developed to globally solve PP(Ω).

A branch-and-bound algorithm [1, 2]:

Step 0: Initialization. Initialize the incumbent solution $x^*=\varnothing$, the incumbent objective value $v^*=\infty$, and the selected active node to be Ω^a = Ω. Solving LP(Ω^a) results in a solution $(\bar{x},\bar{X})$, and the objective value v(LP(Ω^a)) is a lower bound of PP(Ω). If $\bar{x}$ is feasible to PP(Ω), update x^* and v^* if necessary, and if v^* = v(LP(Ω^a)), then stop, and x^* solves PP(Ω). Otherwise, determine a branching variable by a proper branching rule. For details of branching rule, please see [1, 2].

Step 1: Partition. Partition the selected active node into two subhyperrectangles, i.e., two new nodes, by splitting the current bounding interval of the branching variable at some intermediate value.

Step 2: Bounding. For each of the two new nodes, solve the RLT linear programming relaxation problem as done in step 0, and determine a corresponding branching variable by the branching rule for each of the two nodes.

Step 3: Fathoming. Fathom the nonimproving nodes according to some selected optimality tolerance, and update the set of active nodes. If the set of active nodes is empty, then stop. Otherwise, go to step 4.

Step 4: Node selection. Select an active node with the least lower bound, and then return to step 1.

It is proved in [1, 2] that the above algorithm results in a global optimal solution of PP(Ω) within finite branching steps, or if the branching process goes infinitely, then along any infinite branch of the branch-and-bound tree, any accumulation point of the *x*-variable part of the sequence of linear programming relaxation solutions generated for the node subproblems solves PP(Ω).

B. Application of the algorithm to problem (P)

In this section we main introduce the process of applying RLT to problem (P), and the branch-and-bound algorithm can be applied accordingly and will not be described in detail. To apply the RLT based branch-and-bound algorithm, we first transform our model into a polynomial programming form. Let

$$y_{ijt}^{n}=\begin{cases}\dfrac{x_{ijt}^{n}}{JPH_{ijt}} & \text{if } JPH_{ijt}\neq 0\\ 0 & \text{otherwise}\end{cases},\quad y_{ijt}^{o}=\begin{cases}\dfrac{x_{ijt}^{o}}{JPH_{ijt}} & \text{if } JPH_{ijt}\neq 0\\ 0 & \text{otherwise}\end{cases},$$

and then y_{ijt}^{n} (y_{ijt}^{o}, respectively) denotes the normal working (overtime, respectively) hours that distributed to product *j* in plant *i* in period *t*. Substituting y_{ijt}^{n} and y_{ijt}^{o} for corresponding parts in our model, we can equivalently write the formulation as (P1):

$$\min\left[\sum_{i\in I,j\in J,t\in T}sc_{ijt}-\sum_{i\in I,j\in J,t\in T}\left(r_{ijt}^{n}y_{ijt}^{n}JPH_{ijt}+r_{ijt}^{o}y_{ijt}^{o}JPH_{ijt}\right)\right]$$

subject to

$$\begin{cases}JPH_{ijt}\le JPH_{ij0} & \forall i\in I,j\in J,t\in T\\ y_{ijt}^{n}\begin{cases}\le HIW_{it}^{n} & \text{if } JPH_{ijt}\neq 0\\ =0 & \text{otherwise}\end{cases} & \forall i\in I,j\in J,t\in T \quad (10)\\ y_{ijt}^{o}\begin{cases}\le HIW_{it}^{o} & \text{if } JPH_{ijt}\neq 0\\ =0 & \text{otherwise}\end{cases} & \forall i\in I,j\in J,t\in T \quad (11)\\ \sum_{j\in J}y_{ijt}^{n}\le HIW_{it}^{n} & \forall i\in I,t\in T\\ \sum_{j\in J}y_{ijt}^{o}\le HIW_{it}^{o} & \forall i\in I,t\in T\\ \sum_{i\in I}\left(y_{ijt}^{n}JPH_{ijt}+y_{ijt}^{o}JPH_{ijt}\right)\le d_{jt} & \forall j\in J,t\in T\\ JPH_{ijt},y_{ijt}^{n},y_{ijt}^{o}\ge 0, & \forall i\in I,j\in J,t\in T\end{cases}$$

Note that sc_{ijt} is a step function of variable JPH_{ijt}, and constraints (10) and (11) involve logical judgments, which do not correspond to the form of PP(Ω) in section A. Fortunately, it is easy to check that the conclusions of the RLT theory also holds and the branch-and-bound algorithm also works for P1.

The polynomial terms of P1 involve variables including y_{ijt}^{n}, y_{ijt}^{o}, and JPH_{ijt} for $i\in I, j\in J, t\in T$. Let $u_{y_{ijt}^{n}}$, $u_{y_{ijt}^{o}}$, and $u_{JPH_{ijt}}$ ($l_{y_{ijt}^{n}}$, $l_{y_{ijt}^{o}}$ and $l_{JPH_{ijt}}$, respectively) denote the upper (lower, respectively) bound of y_{ijt}^{n}, y_{ijt}^{o} and JPH_{ijt}, respectively, for $i\in I, j\in J, t\in T$ (all variables are larger than 0; if any of the variable has no upper bound, we use a very large constant instead). To reformulate and relax P1, we first write the bound factors for $i\in I, j\in J, t\in T$ as

$$\begin{aligned}&u_{y_{ijt}^{n}}-y_{ijt}^{n}\ge 0,\ y_{ijt}^{n}-l_{y_{ijt}^{n}}\ge 0,\\ &u_{y_{ijt}^{o}}-y_{ijt}^{o}\ge 0,\ y_{ijt}^{o}-l_{y_{ijt}^{o}}\ge 0,\\ &u_{JPH_{ijt}}-JPH_{ijt}\ge 0,\ JPH_{ijt}-l_{JPH_{ijt}}\ge 0,\end{aligned}$$

A large number of implied bound-factor products constraints can be constructed with the above bound factors. However, we only need to add to P1 the constraints below for each *i*, *j*, *t*, which involve polynomial terms that exist in P1:

$$\begin{aligned}&\left(u_{y_{ijt}^{n}}-y_{ijt}^{n}\right)\left(u_{JPH_{ijt}}-JPH_{ijt}\right)\ge 0,\ \left(u_{y_{ijt}^{n}}-y_{ijt}^{n}\right)\left(JPH_{ijt}-l_{JPH_{ijt}}\right)\ge 0,\\ &\left(y_{ijt}^{n}-l_{y_{ijt}^{n}}\right)\left(u_{JPH_{ijt}}-JPH_{ijt}\right)\ge 0,\ \left(y_{ijt}^{n}-l_{y_{ijt}^{n}}\right)\left(JPH_{ijt}-l_{JPH_{ijt}}\right)\ge 0,\\ &\left(u_{y_{ijt}^{o}}-y_{ijt}^{o}\right)\left(u_{JPH_{ijt}}-JPH_{ijt}\right)\ge 0,\ \left(u_{y_{ijt}^{o}}-y_{ijt}^{o}\right)\left(JPH_{ijt}-l_{JPH_{ijt}}\right)\ge 0,\\ &\left(y_{ijt}^{o}-l_{y_{ijt}^{o}}\right)\left(u_{JPH_{ijt}}-JPH_{ijt}\right)\ge 0,\ \left(y_{ijt}^{o}-l_{y_{ijt}^{o}}\right)\left(JPH_{ijt}-l_{JPH_{ijt}}\right)\ge 0,\end{aligned}$$

After reformulation, substitute

$$X_{ijt}^{n}=y_{ijt}^{n}JPH_{ijt},\ X_{ijt}^{o}=y_{ijt}^{o}JPH_{ijt}\ \text{ for } i\in I,j\in J,t\in T$$

and then we obtain the linear programming relaxation problem of P1 as (LP1):

$$\min\left[\sum_{i\in I,j\in J,t\in T}sc_{ijt}-\sum_{i\in I,j\in J,t\in T}\left(r_{ijt}^{n}X_{ijt}^{n}+r_{ijt}^{o}X_{ijt}^{o}\right)\right]$$

subject to

$$\begin{cases}JPH_{ijt}\le JPH_{ij0} & \forall i\in I,j\in J,t\in T\\ X_{ijt}^{n}\le HIW_{it}^{n}JPH_{ijt} & \forall i\in I,j\in J,t\in T\\ X_{ijt}^{o}\le HIW_{it}^{o}JPH_{ijt} & \forall i\in I,j\in J,t\in T\\ \sum_{j\in J}y_{ijt}^{n}\le HIW_{it}^{n} & \forall i\in I,t\in T\\ \sum_{j\in J}y_{ijt}^{o}\le HIW_{it}^{o} & \forall i\in I,t\in T\\ \sum_{i\in I}\left(X_{ijt}^{n}+X_{ijt}^{o}\right)\le d_{jt} & \forall j\in J,t\in T\\ u_{y_{ijt}^{n}}u_{JPH_{ijt}}-u_{y_{ijt}^{n}}JPH_{ijt}-u_{JPH_{ijt}}y_{ijt}^{n}+X_{ijt}^{n}\ge 0, & \forall i\in I,j\in J,t\in T\\ u_{y_{ijt}^{n}}JPH_{ijt}-u_{y_{ijt}^{n}}l_{JPH_{ijt}}-X_{ijt}^{n}+l_{JPH_{ijt}}y_{ijt}^{n}\ge 0, & \forall i\in I,j\in J,t\in T\\ u_{JPH_{ijt}}y_{ijt}^{n}-X_{ijt}^{n}-l_{y_{ijt}^{n}}u_{JPH_{ijt}}+l_{y_{ijt}^{n}}JPH_{ijt}\ge 0, & \forall i\in I,j\in J,t\in T\\ X_{ijt}^{n}-l_{JPH_{ijt}}y_{ijt}^{n}-l_{y_{ijt}^{n}}JPH_{ijt}+l_{y_{ijt}^{n}}l_{JPH_{ijt}}\ge 0, & \forall i\in I,j\in J,t\in T\\ u_{y_{ijt}^{o}}u_{JPH_{ijt}}-u_{y_{ijt}^{o}}JPH_{ijt}-u_{JPH_{ijt}}y_{ijt}^{o}+X_{ijt}^{o}\ge 0, & \forall i\in I,j\in J,t\in T\\ u_{y_{ijt}^{o}}JPH_{ijt}-u_{y_{ijt}^{o}}l_{JPH_{ijt}}-X_{ijt}^{o}+l_{JPH_{ijt}}y_{ijt}^{o}\ge 0, & \forall i\in I,j\in J,t\in T\\ u_{JPH_{ijt}}y_{ijt}^{o}-X_{ijt}^{o}-l_{y_{ijt}^{o}}u_{JPH_{ijt}}+l_{y_{ijt}^{o}}JPH_{ijt}\ge 0, & \forall i\in I,j\in J,t\in T\\ X_{ijt}^{o}-l_{JPH_{ijt}}y_{ijt}^{o}-l_{y_{ijt}^{o}}JPH_{ijt}+l_{y_{ijt}^{o}}l_{JPH_{ijt}}\ge 0, & \forall i\in I,j\in J,t\in T\\ l_{y_{ijt}^{n}}\le y_{ijt}^{n}\le u_{y_{ijt}^{n}}, & \forall i\in I,j\in J,t\in T\\ l_{y_{ijt}^{o}}\le y_{ijt}^{o}\le u_{y_{ijt}^{o}}, & \forall i\in I,j\in J,t\in T\\ l_{JPH_{ijt}}\le JPH_{ijt}\le u_{JPH_{ijt}}, & \forall i\in I,j\in J,t\in T\\ X_{ijt}^{n},X_{ijt}^{o}\ge 0, & \forall i\in I,j\in J,t\in T\end{cases}$$

Here, X^n_{ijt} (X^o_{ijt} , respectively) has the same physical significance with x^n_{ijt} (x^o_{ijt} , respectively), i.e., the production of product j in plant i in period t in normal working (overtime, respectively) hours, but different notations are used to distinguish their different roles. With LP1, we can apply the branch-and-bound algorithm accordingly. In section C we will give a numerical example to demonstration the process of the branch-and-bound algorithm.

C. An illustrative numerical example

In this section we give a numerical example to demonstrate the algorithm described in Section III.A. Assume there are two plants producing two kinds of products in one periods, that is, $I = \{1, 2\}$, $J = \{1, 2\}$, and $T = \{1\}$. Since there is only one period, we omit the index t from the subscripts of the coefficients and variables in the formulation. For simplification, we assume sc_{ij} is proportional to JPH_{ij} with a coefficient scc_{ij} . Note that for the general situations of the piecewise linear setup cost, our model still works. After the RLT procedure, we obtain the linear programming relaxation problem of the example as (LP-e)

$$\min\left[\sum_{i\in I, j\in J} scc_{ij} JPH_{ij} - \sum_{i\in I, j\in J}\left(r^n_{ij} X^n_{ij} + r^o_{ij} X^o_{ij}\right)\right]$$

subject to

$$\begin{cases} JPH_{ij} \le JPH_{ij0} & \forall i\in I, j\in J \\ X^n_{ij} \le HIW^n_i JPH_{ij} & \forall i\in I, j\in J \\ X^o_{ij} \le HIW^o_i JPH_{ij} & \forall i\in I, j\in J \\ \sum_{j\in J} y^n_{ij} \le HIW^n_i & \forall i\in I \\ \sum_{j\in J} y^o_{ij} \le HIW^o_i & \forall i\in I \\ \sum_{i\in I}\left(X^n_{ij} + X^o_{ij}\right) \le d_j & \forall j\in J \\ \text{bound-factor products constraints} & \\ \text{bound constraints for } y^n_{ij},\ y^o_{ij} \text{ and } JPH_{ij}, & \forall i\in I, j\in J \\ X^n_{ij}, X^o_{ij} \ge 0, & \forall i\in I, j\in J \end{cases}$$

We assume the values of the parameters as follows.

$$\left(scc_{ij}\right) = \begin{bmatrix} 2000 & 10000 \\ 10000 & 2000 \end{bmatrix},\ \left(r^n_{ij}\right) = \begin{bmatrix} 20 & 10 \\ 10 & 20 \end{bmatrix},\ \left(r^o_{ij}\right) = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix},$$

$$\left(Z_{ij0}\right) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\ JPH_{ij0} = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix},$$

$$\left[HIW^n_1\ \ HIW^n_2\right] = [120\ 120],\ \left[HIW^o_1\ \ HIW^o_2\right] = [30\ 30]$$

To apply the branch-and-bound algorithm, we set the upper and lower bounds of y^n_{ij}, y^o_{ij} and JPH_{ij} for $i\in I, j\in J$ as

$$\left(u_{y^n_{ij}}\right) = \begin{bmatrix} 120 & 120 \\ 120 & 120 \end{bmatrix},\ \left(l_{y^n_{ij}}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \left(u_{y^o_{ij}}\right) = \begin{bmatrix} 30 & 30 \\ 30 & 30 \end{bmatrix},$$

$$\left(l_{y^o_{ij}}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \left(u_{JPH_{ij}}\right) = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix},\ \left(l_{JPH_{ij}}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Solving LP-e under these bounds[1], the objective value is -28000, which is a lower bound of the primal problem, and the solution is

$$\left(y^n_{ijt}\right) = \begin{bmatrix} 0.0240 & 0 \\ 0 & 0.0240 \end{bmatrix},\ \left(y^o_{ijt}\right) = \begin{bmatrix} 0.0060 & 0 \\ 0 & 0.0060 \end{bmatrix},$$

$$\left(X^n_{ijt}\right) = \begin{bmatrix} 2400 & 0 \\ 0 & 2400 \end{bmatrix},\ \left(X^o_{ijt}\right) = \begin{bmatrix} 600 & 0 \\ 0 & 600 \end{bmatrix},\ \left(JPH_{ij}\right) = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix},$$

which does not satisfy (9) and then branching is needed. We select JPH_{11} to branch the problem into two subproblems with corresponding upper and lower bounds as $0 \le JPH_{11} \le 20$ and $20 \le JPH_{11} \le 100$, respectively (we choose the values of 20 in order to simplify and shorten our example.). Solving the two resulting subproblems, we obtain that their objective values are the same and equal to -28000, which do not update the incumbent lower bound of the primal problem, and that their solution are also the same and do not satisfy (9). Since the objective values of the two subproblems are the same, we now can choose any of them to branch. Here using JPH_{22} we branch the first subproblem (with the constraint $0 \le JPH_{11} \le 20$) into two subproblems with constraints $0 \le JPH_{22} \le 20$ and $20 \le JPH_{22} \le 100$, respectively. Solving the subproblem with constraints $0 \le JPH_{11} \le 20$ and $0 \le JPH_{22} \le 20$, we obtain a solution of

$$\left(y^n_{ijt}\right) = \begin{bmatrix} 120 & 0 \\ 0 & 120 \end{bmatrix},\ \left(y^o_{ijt}\right) = \begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix},$$

$$\left(X^n_{ijt}\right) = \begin{bmatrix} 2400 & 0 \\ 0 & 2400 \end{bmatrix},\ \left(X^o_{ijt}\right) = \begin{bmatrix} 600 & 0 \\ 0 & 600 \end{bmatrix},\ \left(JPH_{ij}\right) = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix},$$

which satisfies (9), and the objective values is -28000, which reaches the lower bound of the primal problem. Then we can claim that this solution is global optimal for the primal problem. The algorithm stops. From the result, it is easy to understand why branching the two variables JPH_{11} and JPH_{22} with value of 20 helps to shorten the branching process. If we adopt other branching values, the process may goes infinite, but we still can obtain gradually smaller subhyperrectangles and the corresponding part of the relaxation solutions will converge to the optimal solution.

IV. A Stochastic Programming Model Concerning Uncertain Demands

In section II and III we proposed a planning model for determined demands, where the production lines in the plants have been built up with given maximal line production rates. In reality, however, we often face uncertain demands. A problem will be how to configure the maximal line production rates of the production lines, that is, how to build the lines, to averagely maximize the profit under uncertain

[1] We use CPLEX to solve the relaxed linear programming problem. CPLEX cannot solve programming problems involving polynomial terms, but can efficiently solve programming problems involving piecewise linear functions, logical constraints or even indicator constraints [5]. So CPLEX can still solve the linear programming relaxation problem if we consider bc_{ij} s and sc_{ijt} s as piecewise linear functions of the corresponding variables.

demands in a long run. We model this problem by modifying (P). Introduce additional given values as:

Z_{ij0} = 1 if it is possible to build a production line of product j in plant i; = 0, otherwise.

bc_{ij} building cost of a production line for product j in plant i, which is considered as a piecewise linear function of JPH_{ij0}.

and consider JPH_{ij0}s for $i \in I, j \in J$ as decision variables, and d_{jt} as random variables, we obtain a modification of (P) as:

$$\max \; -\sum_{i\in I, j\in J} bc_{ij} + \left[\sum_{i\in I, j\in J, t\in T} \left(r_{ijt}^n x_{ijt}^n + r_{ijt}^o x_{ijt}^o \right) - \sum_{i\in I, j\in J, t\in T} sc_{ijt} \right]$$

subject to

$$\begin{cases} JPH_{ij0} \le JPH_{ij0} Z_{ij0} & \forall i \in I, j \in J \\ JPH_{ijt} \le JPH_{ij0} & \forall i \in I, j \in J, t \in T \\ x_{ijt}^n \le HIW_{it}^n JPH_{ijt} & \forall i \in I, j \in J, t \in T \\ x_{ijt}^o \le HIW_{it}^o JPH_{ijt} & \forall i \in I, j \in J, t \in T \\ \sum_{j\in J} \dfrac{x_{ijt}^n}{JPH_{ijt}} \le HIW_{it}^n & \forall i \in I, t \in T, JPH_{ijt} \ne 0 \\ \sum_{j\in J} \dfrac{x_{ijt}^o}{JPH_{ijt}} \le HIW_{it}^o & \forall i \in I, t \in T, JPH_{ijt} \ne 0 \\ \sum_{i\in I} \left(x_{ijt}^n + x_{ijt}^o \right) \le \boldsymbol{d_{jt}} & \forall j \in J, t \in T \\ JPH_{ij0}, JPH_{ijt}, x_{ijt}^n, x_{ijt}^o \ge 0, \;\; \forall i \in I, j \in J, t \in T \end{cases}$$

Denote by $\boldsymbol{JPH}_0$ the decision vector with JPH_{ij0}s for $i \in I, j \in J$ as its elements, by $\boldsymbol{y}$ the decision vector with JPH_{ijt}s, x_{ijt}^n s and x_{ijt}^o s for $i \in I, j \in J, t \in T$ as its elements, and by $\boldsymbol{\xi}$ the random vector with $\boldsymbol{d_{jt}}$ s for $j \in J, t \in T$ as its elements. Let $\boldsymbol{h}(\boldsymbol{JPH}_0, \boldsymbol{y}, \boldsymbol{\xi}) \le \boldsymbol{0}$ represent all the constraints excepts the bound constraints, and

$$c(\boldsymbol{JPH_0}) = -\sum_{i\in I, j\in J} bc_{ij} \,,$$

$$g(\boldsymbol{y}) = \sum_{i\in I, j\in J, t\in T} \left(r_{ijt}^n x_{ijt}^n + r_{ijt}^o x_{ijt}^o \right) - \sum_{i\in I, j\in J, t\in T} sc_{ijt} \,.$$

Follow the nomenclature and formulation methods in [4], we can define the two-stage stochastic program of (P) as

$$\max \; c(\boldsymbol{JPH_0}) + \mathscr{Q}(\boldsymbol{JPH_0})$$

subject to

$$\begin{cases} \boldsymbol{h}(\boldsymbol{JPH}_0, \boldsymbol{y}, \boldsymbol{\xi}) \le \boldsymbol{0}, \\ \boldsymbol{JPH}_0 \ge 0 \end{cases}$$

where

$$\mathscr{Q}(\boldsymbol{JPH_0}) = E_{\boldsymbol{\xi}} Q(\boldsymbol{JPH_0}, \boldsymbol{\xi})$$

$$Q(\boldsymbol{JPH_0}, \boldsymbol{\xi}) = \max_{\boldsymbol{y}} \left\{ g(\boldsymbol{y}) \,\middle|\, \boldsymbol{h}(\boldsymbol{JPH}_0, \boldsymbol{y}, \boldsymbol{\xi}) \le \boldsymbol{0}, \boldsymbol{y} \ge 0 \right\}$$

and $E_{\boldsymbol{\xi}}$ represents the mathematical expectation with respect to $\boldsymbol{\xi}$. In this stochastic programming model, $\boldsymbol{JPH}_0$ is the first stage decision vector, and $\boldsymbol{y}$ is the second stage decision vector. For each product, for example, product j, $\boldsymbol{d_{jt}}$ s for $t \in T$ form a random sequence. In reality, we can estimate the distribution of the random sequences of demands for each product according to the market history and the current market situation using some reasonable model, e.g., Bass model [6]. Although the formulated problem is hard to solve for large scale network in reality, we could approximately solve the two-stage stochastic programming problem by combining Monte Carlo method and the method introduced in section III.

V. Conclusion

We introduced in this paper a general model for manufacturing planning problem, which helps to understand the manufacturing network in reality. The problem is modeled and transformed as a nonlinear polynomial programming problem, for which a RLT based branch-and-bound algorithm is introduced to find the global optimal solution. A numerical example is given to demonstrate the procedure of the algorithm. We also proposed a stochastic programming model under uncertain demands.

References

[1] P.M. Pardalos, H.E. Romeijn, (Eds.), *Handbook of global Optimization*, Vol. 2, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[2] H.D. Sherali, C.H. Tuncbilek, *A global optimization algorithm for polynomial programming problems using a Reformulation-Linearization Technique*, Journal of Global Optimization 2(1): 101-112, 1992.

[3] B. Fleischmann, S. Ferber and P. Henrich, *Strategic Planning of BMW's Global Production Network*, Interfaces 36(3): 194-208, 2006.

[4] J. R. Birge, F. Louveaux, *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

[5] *ILOG CPLEX 11.0 User's Manual*, France, 2007.

[6] F.M. Bass, *A new product growth model for consumer durables.* Management Science 15(5): 215-227, 1969.