

A “Greedy” Czech Manufacturing Case

George Cristian Gruia and Michal Kavan

Abstract—The article describes a case study on one of Czech Republic’s manufacturing middle size enterprises (ME), where due to the European financial crisis, production lines had to be redesigned and optimized in order to minimize the total costs of the production of goods. It is considered an optimization problem of minimizing the total cost of the work load, according to the costs of the possible locations of the workplaces, with an application of the Greedy algorithm and a partial analogy to a Set Packing Problem. The displacement of working tables in a company should be as a one-to-one monotone increasing function in order for the total costs of production of the goods to be at minimum. We use a heuristic approach with greedy algorithm for solving this linear optimization problem, regardless the possible greediness which may appear and we apply it in a Czech ME.

Keywords—Czech, greedy algorithm, minimize, total costs.

I. INTRODUCTION

THE impetuous for our research comes from a real problem that manufacturers are facing as a result of the economic and financial crisis, which is far from over in several European countries like Czech Republic, in our case. In the manufacturing industry, due to today’s market conditions, when the production begins, it begins also the loss of working time and increase in costs, which is an important barrier for an increase of the degree of productivity, as it showed us concurrent manufacturing companies in countries like Germany, South Korea, USA or Japan. In these economic frustrating times for investors to decide where or in which companies they should put their money, manufacturing companies, which are responsible for the long term increase in national gross product, should try to solve their projects by optimization of the manufacturing lines. There are several ways how management can improve the productivity and efficiency of the process like Lean, Six Sigma or by simply applying tools like 5S or Root cause analysis, but everything should be done in time and what was valid 30 years ago it doesn’t always apply in today’s market.

Today, the manufacturing process is managed according to the customers’ requirements and in order to fulfill these requirements, every process, every job must be done in time, maintaining the required quality level, projects being in 99.99

This paper has been supported by the Czech Science Foundation GA ČR in the frame of GA ČR P403/12/1950 project “Design of Experiments for Product Development and Multi-factor Optimization of Production” by the Ministry of Education, Youth and Sports of the Czech Republic.

George C. Gruia is with the Management and Economics Department, Faculty of Mechanical Engineering, Czech Technical University in Prague (phone: +42(0)776-211-483; e-mail: GeorgeCristian.Gruia@fs.cvut.cz).

Michal Kavan is with the Management and Economics Department, Faculty of Mechanical Engineering, Czech Technical University in Prague (phone: +42(0)224-355-795; e-mail: Michal.Kavan@fs.cvut.cz).

% of the cases, time constrained projects. The Project Manager should solve problems and implement projects with limited and adjacent resources in a predefined time period. Adjacent resources are resources for which the units assigned to a job are required to be in some sense adjacent. Possible examples of adjacent resources are dry docks, shop floor spaces and assembly areas. Otherwise, costs will rise and the company will not be able to remain long on the competitive market.

There are several studies regarding the Time-Constrained Project Scheduling problem (TCPS) with one adjacent 1-dimensional or 2-dimensional adjacent resource, ([1], [2] with one 1-dimensional adjacent resource), but we decided to focus our study on a more general aspect of the problem, i.e., the optimization of manufacturing time by minimizing the total input costs from the point of view of distributing the work load on the available production lines, in order to answer the actual market conditions.

The present paper is structured in the following way: first it is presented the problem definition from a mathematical and theoretical point of view and then a practical part is presented where it was used data from a Czech middle size manufacturing company and computed using Cplex Optimization Studio program [3]. At the end results are interpreted, conclusions are drawn based on the practical part and future lines are drawn for future studies.

II. PROBLEM DEFINITION

In a manufacturing company, in order to produce goods we should have within a predefined time frame, a defined number of operations, given by a set of jobs, a set of renewable resources and one or multiple 1-dimensional adjacent resources, which will transform the inputs into outputs ready to be sold. Every job should be done within a specific time and space frame in order for the production to be Just-in-time. We consider that the time is determined by the work distribution on the production line, so only the distribution of the work load is considered below. Time spent with transport from the factory to the client is not considered.

In a market where time is money, minimizing the total input costs will minimize the manufacturing time of the products and the customers will receive their ordered goods in time, without additional delays.

Every manager confronts with the following problem, which we’ve decided to study further in our present paper:

In what way should the work load (jobs or/and machines) be divided on the production line, in order to minimize the total costs of the work, which should be done, in order for the goods to be produced? in order to be competitive and fulfill

the market requirements in time, possible with an extra time reserve.

In this manner we tried to give answers to the above stated question by solving it from a greedy perspective. We assume that the work, which has to be done, is executed not only by specialized workers (or robots or automatic machines) but also by workers (or robots or automatic machines) who know to do several jobs according to their qualifications and on the manufacturing line perform several operations, being able to switch to other working tables. Or in case of the automatic production lines, robots perform several operations according to their input programmed code.

A. Theoretical Part

If a good in order to be manufactured and send to the customers, must pass through several production stages and operations. According to the type and complexity of the finite product, the production managers together with the HR department or the IT department, should solve greater problems of dividing the jobs between workers and machines/robots with the aim in minimizing the total cost of the work. The car manufacturing industry is one of the most complex manufacturing industries, where our study was implemented. Automatizing the entire company would be the expensive answer for our research question; however without the partial automatization, at least, in today's manufacturing industry, a company cannot succeed. Thus, we look at this problem from the point of view of optimizing the total input costs, by dividing the work load between workers and automated machines on the available production lines.

Given a set of integer values, representing the work (jobs) to be done, in a manufacturing company, $\{j_1, j_2, \dots, j_n\}$, where j_1 is the 1st job, which has to be done at the entry of the material in the company, and states the first work place from where we can start the work, j_n = last job and states the n^{th} place where we can move the working table for performing the job to minimize the input costs, and $j_i < j_{i+1}$ for $1 \leq i \leq n-1$, and a positive integer J , find a set of non-negative integers $\{c_1, c_2, \dots, c_n\}$ which minimize $\sum_{i=1}^n c_i$ subject to $\sum_{i=1}^n c_i j_i$.

The finite product has several partial jobs, which should be finished by workers or/and machines on a work alone basis, before passing to the next work place (work phase). The following pictures show some abstract examples of possible manufacturing lines. In these pictures the Inputs are described as "1111" and Outputs as "0000". In other words as soon as the product is partially worked (semi-finished) the 1 is transformed to 0 and passes to another working table.

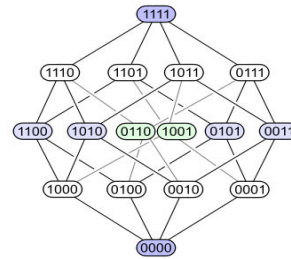


Fig. 1 A 4-dimensional hypercube [4]

These pictures are also called Hasse diagrams, after Helmut Hasse (1898–1979); according to Birkhoff [5], they are so-called because of the effective use Hasse made of them. However, Hasse was not the first to use these diagrams; they appear, for e.g., in Vogt [6]. Although Hasse diagrams were originally devised as a technique for making drawings of partially ordered sets by hand, they have more recently been created automatically using graph drawing techniques [7], [8].

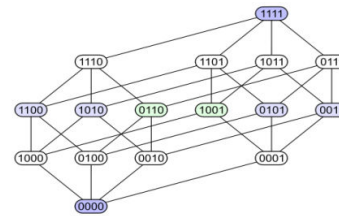


Fig. 2 A pseudo-projection of the 4-dimensional hypercube [4]

We consider the situation when an intermediate production phase is critical to the finalization of the good. Thus if an intermediate production phase is not fulfilled within product's requirements, the whole production line will be interrupted and the semi-finished good will never get into the customers' hands. This situation is typical for companies which implemented Lean Manufacturing and use Andon signals, the line being stopped when a fault is discovered.

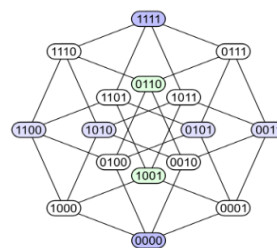


Fig. 3 The edge-first-shadow of the 4-dimensional hypercube [4]

In order to solve our problem we translate it in a linear optimization problem which we can solve using the Greedy algorithm. Before solving the optimization problem we must define that the intermediate operations are critical to the quality of the final product. Thus we first examine the structure of the distributive lattice (formed of the working tables within the production line) $D \subseteq 2^E$ and show the one-

to-one correspondence between the set of distributive lattices $D \subseteq 2^E$ with $\emptyset, E \in D$ and the set of partially ordered sets (also called posets) on partitions of E.

For a distributive lattice $D \subseteq 2^E$ the cardinality $|D|$ of D can be as large as 2^E and listing all the elements of D to represent it is not practical even for medium-sized E. We shall show how to efficiently express a distributive lattice as a structured system, a poset, on E. In other words, by using the greedy algorithm we find the efficient distribution of work load within a company, function of the individual costs, c, which are noted as x in the following equations and X is the total costs of the work load.

Let $D \subseteq 2^E$ be a distributive lattice with $\emptyset, E \in D$. A sequence of monotone increasing elements of D

$$C: V_0 \subset V_1 \subset V_2 \subset \dots \subset V_k \quad (1)$$

is called a chain of D and k is the length of the chain C. If there exists no chain which contains chain C as a proper subsequence, C is called a maximal chain of D. And if C given by (1) is a maximal chain, we have $V_0 = \emptyset$ and $V_k = E$.

For each $e \in E$ we define

$$D(e) = \bigcap \{X \mid e \in X \in D\} \quad (2)$$

$D(e)$ is the unique minimal element of D, which contains e.

Thus we can state that for

$$(\forall) e \in E \text{ and } e^i \in D(e), (\exists) D(e^i) \subseteq D(e) \quad (3)$$

Also we consider $G(D) = (E, R(D))$ be a directed graph with vertex set E and arc set $R(D)$, where

$$R(D) = \{(e, e^i) \mid e \in E, e^i \in D(e)\} \quad (4)$$

We decompose the graph $G(D)$ into connected components $G_i = (S_i, R_i)$, where $i \in I$. Let \leq_D be the partial order on the set of the connected components $\{G_i \mid i \in I\}$ naturally induced by the decomposition, that is $G_{i_1} \leq_D G_{i_2}$, where $i_1, i_2 \in I$ if and only if exists a directed path from a vertex of G_{i_2} to a vertex of G_{i_1} . But according to (3) $G(D)$ is transitively closed, thus if there is a directed path from a vertex v_1 to a vertex v_2 , then there is an arc (v_1, v_2) in $G(D)$. Therefore, we are sure when we state that if $G_{i_1} \leq_D G_{i_2}$, there exists an arc from any vertex of G_{i_1} to any vertex of G_{i_2} .

Denote the set of the vertex sets $S_i (i \in I)$ of the connected components $G_i (i \in I)$ by

$$\Sigma(D) = \{S_i \mid i \in I\} \quad (5)$$

where $\Sigma(D)$ is a partition of E. In the following we consider \leq_D as a partial order on $\Sigma(D)$ by identifying G_i with S_i for each $i \in I$. We have thus obtained a poset $P(D) = (\Sigma(D), \leq_D)$, which is called the poset derived from distributive lattice D.

For a poset $P = (P, \leq)$ a set $L \subseteq P$ is called a lower ideal of P, if each $e, e^i \in P$ we have

$$e \leq e^i \in L \Rightarrow e \in L \quad (6)$$

Birkhoff [7] introduced the **theorem (1)** that: Let $D \subseteq 2^E$ be a distributive lattice with $\emptyset, E \in D$. Then, for the poset $P(D) = (\Sigma(D), \leq_D)$ derived from D, the following (i) and (ii) hold:

(i) For each ideal L of $P(D)$,

$$\bigcup \{F \mid F \in L\} \in D \quad (7)$$

(ii) For each $X \in D$, there exists an ideal L of $P(D)$ such that

$$X = \bigcup \{F \mid F \in L\} \quad (8)$$

Proof: For (i): Put $X = \bigcup \{F \mid F \in L\}$. Since L is an ideal of $P(D)$, it allows from the definition of $P(D)$ that $D(e) \subseteq X$ for each $e \in X$. Then $X = \bigcup \{D(e) \mid e \in X\}$ and we have $X \in D$ since $D(e) \in D$ from (2).

For (ii): For a given $X \in D$, X and any $F \in \Sigma(D)$ do not cross, i.e., either $F \subseteq X$ or $F \subseteq E - X$ due to the partition of $P(D)$. Therefore, $L = \bigcup \{F \mid F \in \Sigma(D), F \subseteq X\}$ is a partition of X. Moreover, because of the definition of $P(D)$, $F_1 \leq_D F_2 \subseteq X$ implies $F_1 \subseteq X$. Consequently, $L = \{F \mid F \in \Sigma(D), F \subseteq X\}$ is a desired ideal of $P(D)$. (q.e.d.)

From theorem (1), with (7) or (8), we determine a one-to-one correspondence between D and the set of all the ideals of $P(D)$.

Given a distributive lattice $D \subseteq 2^E$ with $\emptyset, E \in D$ let

$$C: \emptyset = V_0 \subset V_1 \subset V_2 \subset \dots \subset V_k = E \quad (9)$$

Be an arbitrary maximal chain of D. Then we have

$$\Sigma(D) = \{V_i - V_{i-1} \mid i = 1, 2, \dots, k\} \quad (10)$$

In particular, the length of any maximal chain of D is independent of the choice of a maximal chain and is equal to $|\Sigma(D)|$. We call D simple if the partition $\Sigma(D)$ is composed of singletons of E alone, i.e. $\Sigma(D) = \{\{e\} \mid e \in E\}$.

A sub-modular system (D, f) with simple D is called simple. For a non-simple sub-modular system (D, f) on E, we define

$$\hat{X} = \{F \mid F \in \Sigma(D), F \subseteq X\} \quad (X \in D) \quad (11)$$

$$\hat{D} = \{\hat{X} \mid X \in D\} \quad (12)$$

$$\widehat{f}(\hat{X}) = f(X) \quad (X \in D) \quad (13)$$

Then we have a simple sub-modular system (\hat{D}, \widehat{f}) on $\Sigma(D)$ which we call the simplification of (D, f) .

Now returning to our initial research question, for a sub-modular system (D, f) on E we consider a linear optimization problem described as follows:

$$P_j: \text{minimize } \sum_{e \in E} j(e)c(e) \quad (14)$$

$$\text{Subject to } c \in B(f) \quad (15)$$

where $j: E \rightarrow R$ is a given weight function (individual jobs) and c are individual costs related to the individual jobs which are required to be done for goods to be produced. We assume that (14) is well defined. An optimal solution of P_j is called a minimum-weight base of (D, f) with respect to the weight function j . Similarly, a maximum-weight base of (D, f) with respect to the weight function j is an optimal solution of problem P_j with the weight function $-j$.

If the jobs are arranged as shown above in different types of polyhedron, we can consider the correspondent costs arranged in a similar way. Thus fundamental structural properties of the base polyhedron $B(f)$ are given by the following theorems.

When (D, f) is not simple, problem P_j does not have a finite optimal solution if $j(e) \neq j(e^i)$ for any $e, e^i \in F \in \Sigma(D)$. Therefore, if P_j has an optimal solution, $j: E \rightarrow R$ is constant on each $F \in \Sigma(D)$, and hence it suffices to consider the simplification of (D, f) .

We suppose without loss of generality, that in the minimum-weight base problem P_j described by (14) and (15) $B(f)$ is pointed, i.e., $D = 2^P$ with $P = (E, \leq)$.

Theorem 2[9]: Problem P_j in (15) has a finite optimal solution if and only if $j: E \rightarrow R$ is a monotone non-decreasing function from $P = (E, \leq)$ to (R, \leq) , i.e., $(\forall) e, e^i \in E: e \leq e^i \Rightarrow j(e) \leq j(e^i)$.

Proof: "IF" part: Suppose that j is a monotone non-decreasing function from $P = (E, \leq)$ to (R, \leq) and that the distinct values of weights $j(e)$ ($e \in E$) are given by

$$j_1 < j_2 < \dots < j_p \quad (16)$$

Define

$$A_i = \{e \mid e \in E, j(e) \leq j_i\} \text{ where } i = 1, 2, \dots, p \quad (17)$$

wherenote that $A_p = E$. The sets A_i form a chain $A_1 \subset A_2 \subset \dots \subset A_p$ of D . We recall x =individual job costs.

Lemma 1: For any $A \in D$ let x^A be a base of the reduction $(D, f) \cdot A$ of submodular system (D, f) to A and x_A be a base of the contraction $(D, f) / A$ of (D, f) by A . Then the direct sum $\hat{x} = x^A \oplus x_A$ of x^A and x_A defined by

$$(x^A \oplus x_A)(e) = \begin{cases} x^A(e), & \text{for } e \in A \\ x_A(e), & \text{for } e \in E - A \end{cases} \quad (18)$$

is a base of (D, f) .

From Lemma 1 there exists a base $x \in B(f)$ such that $x(A_i) =$

$$f(A_i), \text{ for } i=1, 2, \dots, p. \quad (19)$$

Then for any base $y \in B(f)$ we have from (15), (16) and (18) the following:

$$\begin{aligned} \sum_{e \in E} j(e)y(e) - \sum_{e \in E} j(e)x(e) &= \sum_{i=1}^p \sum_{e \in A_i - A_{i-1}} j_i(y(e) - x(e)) \\ &= \sum_{i=1}^p \{j_i(y(A_i) - x(A_i)) - j_i(y(A_{i-1}) - x(A_{i-1}))\} \\ &= \sum_{i=1}^{p-1} (j_i - j_{i+1})(y(A_i) - x(A_i)) + j_p(y(A_p) - x(A_p)) \\ &= \sum_{i=1}^{p-1} (j_{i+1} - j_i)(f(A_i) - y(A_i)) \geq 0 \end{aligned} \quad (20)$$

where we define $A_0 = \emptyset$ and we recall $A_p = E$. This shows the optimality of x (which we initially denoted the costs).

"ONLY IF" part: Suppose that j is not a monotone non-decreasing function from $P = (E, \leq)$ to (R, \leq) , i.e., for some k ($1 \leq k < p$), A_k defined by (16) does not belong to D . Then there exist elements $e \in A_k$ and $e^i \in E - A_k$ such that for $(\forall) X \in D$ with $e \in X$ we have $e^i \in X$. Hence, for any base $x \in B(f)$ we have $\tilde{c}(x, e, e^i) = +\infty$ and $x + \alpha(\chi_e - \chi_{e^i}) \in B(f)$ for any $\alpha > 0$. Since $j(e) < j(e^i)$, problem P_j does not have a finite optimal solution. (q.e.d.)

Corollary: Let P_j' be the problem given by P_j where $B(f)$ is replaced by $P(f)$. Problem P_j' has a finite optimal solution if and only if $j: E \rightarrow R$ is a non-positive monotone non-decreasing function from $P = (E, \leq)$ to (R, \leq) .

Proof: The non-positiveness is imposed on j since for any $x \in P(f)$ and $e \in E$ we have $x - \alpha \chi_e \in P(f)$ for an arbitrary $\alpha \in R_+$. For any non-positive j , P_j' has a finite optimal solution if and only if P_j has a finite optimal solution. Therefore, the present corollary follows from theorem 2. (q.e.d.)

Theorem 3: Suppose that j is a monotone non-decreasing function from $P = (E, \leq)$ to (R, \leq) , i.e., the sets A_i , where $i=1, 2, \dots, p$, defined by (16) form a chain of D . For each $i=1, 2, \dots, p$ let f_i be the rank function of the set minor $(D, f) \cdot A_i / A_{i-1}$, where $A_0 = \emptyset$. Then the set of all the optimal solutions of problem P_j is given by

$$\begin{aligned} B(f_1) \oplus B(f_2) \oplus \dots \oplus B(f_p) &= \\ = \{x_1 \oplus x_2 \oplus \dots \oplus x_p \mid x_i \in B(f_i), i = 1, 2, \dots, p\} \end{aligned} \quad (21)$$

wherethe direct sum \oplus is defined by the following: $x \in B(f)$ is an optimal solution of P_j if and only if x restricted on $A_i - A_{i-1}$ is a base of $(D, f) \cdot A_i / A_{i-1}$ for each $i=1, 2, \dots, p$.

Proof: It follows from the proof of the "IF" part of theorem 2 that $x \in B(f_1) \oplus B(f_2) \oplus \dots \oplus B(f_p)$ is an optimal solution of problem P_j . On the other hand, if x is an optimal solution, then we must have a dependency $dep(x, e) \subseteq A_i$ for each $i=1, 2, \dots, p$ and $e \in A_i$.

Theorem 4: A base $x \in B(f)$ is an optimal solution of problem P_j if and only if for each $e, e^i \in E$ such that $e^i \in dep(x, e)$ we have

$$j(e) \geq j(e^i) \quad (22)$$

Proof: The “ONLY IF” is trivial. The “IF” part follows from theorem 3. For, if (22) holds for each $e, e^i \in E$ such that $e^i \in \text{dep}(x, e)$, then we have (19) for A_i , $i=1,2,\dots,p$, defined by (16). (q.e.d.)

Theorem 4 states that the local optimality with respect to elementary transformations from x to $x+\alpha(\chi_e - \chi_{e^i})$, ($e \in E, e^i \in \text{dep}(x, e), \alpha \geq 0$) implies the global optimality.

A sequence (e_1, e_2, \dots, e_n) of all the elements of E (a linear of total ordering of E) is called a linear extension of

$$P = (E, \preceq) \text{ if } e_i \preceq e_j \text{ implies } i \leq j \ (i, j = 1, 2, \dots, n).$$

Furthermore, a linear extension (e_1, e_2, \dots, e_n) of $P = (E, \preceq)$ is called monotone non-decreasing with respect to $j: E \rightarrow R$ if $j(e_1) \leq j(e_2) \leq \dots \leq j(e_n)$. Such a monotone non-decreasing linear extension of $P = (E, \preceq)$ exists if and only if j is a monotone non-decreasing function from $P = (E, \preceq)$ to (R, \leq) . Suppose we are given a monotone non-decreasing weight function j from $P = (E, \preceq)$ to (R, \leq) .

We apply the Greedy algorithm in 2 steps:

1. Find a monotone non-decreasing linear extension (e_1, e_2, \dots, e_n) of $P = (E, \preceq)$ with respect to j .
2. Compute a vector $x \in R^E$ by

$$x(e_i) = f(S_i) - f(S_{i-1}), \text{ where } i = 1, 2, \dots, n \quad (23)$$

where for each $i=1,2,\dots,n$, S_i is the set of the first i elements of (e_1, e_2, \dots, e_n) and $S_0 = \emptyset$. Then x is a minimum-weight base of (D, f) with respect to weight w .

Due to theorem 3, every extreme minimum-weight base can be obtained by the greedy algorithm by appropriately choosing a monotone non-decreasing linear extension (e_1, e_2, \dots, e_n) of D in the 1st step.

The greedy algorithm is not the only solution for minimizing the total costs of the individual jobs, according to their location on the production line, which have to be done in order to produce a finite good. However it gives us a chance of optimizing the work load, for a generally considered manufacturing company. We can conclude that for solving the P_j problem, the solution is a unique optimal one if $j: E \rightarrow R$ is a one-to-one monotone increasing function from $P = (E, \preceq)$ to (R, \leq) .

Theorem 5: Let $f: D \rightarrow R$ be a function on a simple distributive lattice $D \subseteq 2^E$ such that $\emptyset, E \in D$ and $f(\emptyset) = 0$. Then, the greedy algorithm described above works for $B(f)$ defined as follows:

$$B(f) = \{x \mid x \in R^E, (\forall) X \in D: x(X) \leq f(X), x(E) = f(E)\} \quad (24)$$

if and only if f is a sub-modular function of D .

Proof: It suffices to show the “ONLY IF” part. Suppose that the greedy algorithm works for $B(f)$. Then, for each maximal chain

$$C: \emptyset = V_0 \subset V_1 \subset V_2 \subset \dots \subset V_n = E \quad (25)$$

of D the vector $x \in R^E$ defined by (23) belongs to $B(f)$. For any incomparable $X, Y \in D$ choose a maximal chain C of (25) containing $X \cap Y$ and $X \cup Y$ and define x by (23). Since $x \in B(f)$, by definition we have:

$$\begin{aligned} x(X) &\leq f(X), x(Y) \leq f(Y), \\ x(X \cup Y) &= f(X \cup Y), x(X \cap Y) = f(X \cap Y) \end{aligned} \quad (26)$$

Hence we have

$$\begin{aligned} f(X) + f(Y) &\geq x(X) + x(Y), \\ f(X) + f(Y) &= x(X \cup Y) + x(X \cap Y) = \\ &= f(X \cup Y) + f(X \cap Y) \end{aligned} \quad (27)$$

It follows that function “ f ” is a sub-modular function of D . (q.e.d.)

B. Practical Part

We’ve implemented the above stated algorithm in a simplified form, in a manufacturing Czech company from the car manufacturing industry where a certain level of quality is required and the relationship and cooperation between this company and others from car manufacturing industry should be at the highest level in order to answer the market requirements.

An optimization problem was solved, by minimizing the total costs of the work by distributing the working tables in a certain order, according to the Hasse diagram described above. A lattice of 24 was used according to the greedy algorithm from above. Cplex Optimization Studio was used.

Our problem can also be considered as a special case of SPP (set packing problem), a classical optimization problem, one of Karp’s 21 NP-hard problems [10]. A lot of attention is given in specialized literature to set covering problem and node packing problems. For additional information we refer to [11], [12], for set covering problem or [13] and [14], for node packing problem. A good method for solving this type of problem is a Branch and Cut algorithm using polyhedral theory to obtain facets, by determining cliques as defined by [15]. However, only small-sized instances can be exactly solved.

The considered Czech company, further “the Company”, is an ISO-9001 and ISO-TS 16949 certified company that is specialized in bending, forming and assembly of tubes in 2D and 3D. The Company processes steel, stainless steel, copper and aluminum tubes. It is also possible to assemble the products through welding and brazing. Their strength lies in the production of small to medium series, the average diameter of the tube between 6mm-65mm away. Their metal workings are used by leading companies (OEMs) in the automotive, medical, heating and cooling industry. They also offer various finishing’s such as galvanizing or powder coating.

The Company wants to introduce a new innovative product P1 on the market, which will incorporate the latest

technologies and know-how of the company, but for this a production line should be developed where a number of 10 machines should be placed on some of the 15 available production lines in the company. Each of these lines has corresponding costs according to the place of the machines (or working tables for workers). These costs can include different types of operational costs, costs with trained workers, material, up to costs per m2, special tools designed for the production, etc.

Company's target is to implement the new production line and to arrange the machines accordingly to minimize the total input costs. From the table below one can see in what way the machines were displaced, 1 is for a place occupied by the machine and 0 is for an "empty" space on the production line.

We know the capacity of the possible production lines as follows:

TABLE I
CAPACITY AND FIXED COSTS FOR EACH OF THE AVAILABLE PRODUCTION LINES

Lines	Capacity [x10 ³ Nh]	Fixed costs [x10 ³ CZK]
L1	4	200
L2	3	130
L3	3	150
L4	3	250
L5	3	280
L6	2	480
L7	2	200
L8	2	320
L9	2	340
L10	2	300
L11	2	100
L12	1	110
L13	1	120
L14	1	130
L15	1	140

Fixed costs of the lines were computed according to the possible positions of the machine/working table. These positions have also different operational costs related to the distance to the supply room, energy, heat, time spent to get from one point to another by designated workers, fuel consumption of manipulation machines, facility costs.

We also know that some operations are more complex than others and thus the machines are displaced in such a manner.

We use the following formula:

$$TC = FC + VC \text{ (Total Costs = Fixed Costs + Variable Costs)}$$

$$TC = FC \text{ of the opened line} \times \text{the lines which we should open} \\ + \text{costs related to the location of each machine on the line} \times \\ \text{the selected locations of the machines}$$

A simplified version of the initial problem is: Where and on which lines should we locate our machines in order to minimize the total input costs of the production line?

This we can translate in a simplified greedy algorithm (for variable costs):

$$VC := 0$$

$$\text{For } i=1 \text{ to } 15 \text{ and } j=1 \text{ to } 10$$

repeat

$$VC_1 := \text{greedySolution}(\sum_{i=1}^{15} c_i x_i)$$

$$VC_2 := \text{localSearch}(\text{greedySolution})$$

$$VC := VC_2$$

$$\text{until } VC = \sum_{i=1}^{15} c_i x_i = \min$$

where x is the location of each machine (can be either 1 or 0) and is the variable and c_i is the costs of operations (jobs) which had to be made according to the following equation, not to exceed the total capacity of the company:

$$4x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + 2x_6 + 2x_7 + 2x_8 + \\ + 2x_9 + 2x_{10} + 2x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \\ = \text{total capacity of the company}$$

After finding the minimum variable costs we compute the fixed costs according to the specified location.

The costs for each location in our 10x15 matrix of possible locations are:

TABLE II
COSTS OF EACH AVAILABLE LOCATION ON THE PRODUCTION LINES

Machines or Resources "I"	Costs of machine "I" on line "L" [x10 ³ CZK]														
r1	73	2	70	4	5	6	7	8	9	10	11	2	6	56	23
r2	2	3	4	5	6	1	8	9	32	2	12	35	9	2	3
r3	3	4	7	47	7	3	9	10	11	7	13	7	9	87	12
r4	4	5	76	7	63	9	10	4	12	50	89	4	77	98	2
r5	24	6	26	8	35	10	9	12	13	14	15	1	5	3	5
r6	6	7	8	22	10	47	12	13	75	15	16	14	22	64	2
r7	7	12	9	10	11	12	13	14	1	16	17	11	42	72	13
r8	8	9	8	1	12	1	14	15	16	1	18	2	27	5	34
r9	14	10	11	12	1	14	235	1	12	18	86	1	34	16	14
r10	10	1	12	45	14	15	16	17	1	19	1	3	31	29	5

III. RESULTS

A result of 1.564 mil. CZK was obtained for the production line of P1, which was the optimal minimal solution. It was computed a number of 53 iterations and the machines were displaced according to the maximum capacity of the line and of the company. Also a distribution of the working tables was used according to the above mentioned Hasse diagram (Fig.1), which complicated the results. If it wasn't for the Cplex software, the computation of such a relatively easy problem would have taken a lot of time.

The following results were obtained:

TABLE III
LOCATION OF RESOURCE I ON THE PRODUCTION LINE L

Resources	Location of resources "I" on each line "L"														
r1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
r2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
r3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r4	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0
r5	1	1	0	1	0	0	1	0	0	0	1	1	0	0	0
r6	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
r7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
r8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
r9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
r10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

TABLE IV
A COMPARISON OF THE ACTUAL AND AVAILABLE CAPACITY

No. of machines supported by a production line	Capacity
4	4
3	3
3	3
3	3
3	3
0	0
2	2
0	0
0	0
2	2
1	1
0	0
0	0
0	0

According to the Table III, on the fourth line we have 1 on 1st, 2nd, 4th and 7th column which can represent either only a 25% capacity use of the 4th resource (machine and/or workers) or an operation formed of 4 different smaller operations which are interconnected and in cooperation. This was stated as an input constraint according to the complexity of the 4th operation which will take place and also in order to have a Hasse distribution of working load diagram.

One can see that on each line we have at least one machine (working table), so the problem is solved. Also the capacity of the company is maintained and the total costs are minimized by not opening the 6th, 8th, 9th, 10th, 13th, 14th and 15th available production lines, where machines and working table could have been located. Thus the management decided to implement a production line where machines (and working tables) would have been located in different places of the available space, instead of picking one of the available production lines, in order to minimize the total input costs of their new product P1.

Due to the intellectual property rights the data used in this paper were altered not to affect the business development of the company.

IV. CONCLUSION

In this paper we have considered a partial analogy to a special case of the classic combinatorial optimization problem SPP, which was solved with the help of optimization program Cplex, but which is a NP hard problem and cannot be easily solved for larger variables.

We wanted to show that a general problem can be solved with the use of greedy algorithm and linear programming, when applied for a particular case, within the car manufacturing industry as considered above.

REFERENCES

- [1] Guldmond T.A., Hurink J.L., Paulus J.J., Schutten J.M.J., "Time-constrained project scheduling" In *Journal of Scheduling* 2008; 11(2):137–48.
- [2] Hurink J.L., Kok A.L., Paulus J.J., Schutten J.M.J., "Time-constrained project scheduling with adjacent resources" In *Journal of Computers & Operations Research* 2011; 38:310–319.
- [3] IBM ILOG CPLEX Optimization Studio v.12.5
- [4] Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Hasse_diagram
- [5] Birkhoff, Garrett, *Lattice Theory* (Revised ed.), American Mathematical Society, 1948.
- [6] Vogt, Henri Gustav, *Leçon sur la résolution algébrique des équations*, Nony, p. 91, 1895.
- [7] Di Battista, G., Tamassia, R., "Algorithms for plane representation of acyclic digraphs" In *Journal of Theoretical Computer Science* 1988; 61(2-3):175–178, doi:10.1016/0304-3975(88)90123-5.
- [8] Birkhoff, Garrett, *Lattice Theory*, American Mathematical Colloquium Publications 1967:25, 3rd ed., Providence, R. I.
- [9] Fujishige, S., Tomizawa, N., "A note on sub-modular functions on distributive lattices" In *Journal of the Operations Research Society of Japan* 1983; 26:309-318.
- [10] Karp, R. M., "Reducibility Among Combinatorial Problems" In R. E. Miller and J. W. Thatcher (editors). *Complexity of Computer Computations*, New York: Plenum, 1972: 85–103.
- [11] Bianchi S., Nasini G., Tolomei P., "The set covering problem on circulant matrices: polynomial instances and the relation with the dominating set problem on webs" *Electronic Notes in Discrete Mathematics* 2010; 36:1185–1192.
- [12] Zahra Naji-Azimi, Paolo Toth, Laura Galli, "An electromagnetism metaheuristic for the unicast set covering problem" In *European Journal of Operational Research* 2010; 205:290–300.
- [13] Peter J. Zwaneveld, Leo G. Kroon, Stan P.M. van Hoesel, "Routing trains through a railway station based on a node packing model" In *European Journal of Operational Research* 2001; 128:14-33.
- [14] Yuma Tanaka, Shinji Imahori, Mihiro Sasaki, Mutsunori Yagiura, "An LP-based heuristic algorithm for the node capacitated in-tree packing problem" In *Journal of Computers & Operations Research* 2012; 39:637–646.
- [15] M.W. Padberg, "On the facial structure of set packing polyhedral" In *Mathematical Programming* 1973; 5:199–215.



George Cristian Gruia, born in 1984 is a PhD. candidate at Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Management and Economics. His major is production and operations management with a focus on the quality management.

He teaches subjects like Statistics, Project Management and Management and Economics of the Enterprise. He had participated at several international conferences where has received several awards for his papers. He had received a scholarship for complete university studies from the Government of Czech Republic and collaborates with different Czech and Romanian companies.

Michal Kavan, born in 1953 has graduated from the Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Management and Economics. At this university he has been teaching



for more than thirty years, now as an associate professor. He lectures on a number of management courses in master's and doctoral studies. Its basic specialization is production and operations management.

During his professional career he also served as a member of the board of directors of several engineering companies, where he was involved in their management. He graduated a four-week instructor's training course at Saint Mary's University in Halifax, sponsored by the Government of Canada. He has written a number of textbooks and teaching materials.