PRODUCT MIX OPTIMIZATION FOR A SEMICONDUCTOR FAB: MODELING APPROACHES AND DECOMPOSITION TECHNIQUES

Andreas Klemmt

Infineon Technologies
Königsbrücker Straße 180
01099 Dresden, GERMANY

Martin Romauch

University of Vienna
Brünner Straße 72
1220 Vienna, AUSTRIA

Walter Laure

Infineon Technologies
Siemenstraße 2
9500 Villach, AUSTRIA

ABSTRACT

For optimizing a semiconductor fab we are aiming to match the production capabilities and capacities with the demand in the most profitable way. In this paper we address a linear programming model of the product mix problem considering product dependent demand limits (obligations and demand forecast) and profits while respecting the capacity bounds of the production facility. Since the capacity consumption is highly depended on choosing from different production alternatives we are implicitly solving a static capacity planning problem for each product mix. This kind of planning approach is supported by the fluid flow concept of complete resource pooling in high traffic. We propose a general model that considers a wide range of objectives and we introduce a decomposition heuristic. The computational study of the approaches is based on real world data and on randomly generated instances.

1 INTRODUCTION

In this paper we are considering a fab with certain capabilities and capacities (technology and scale of the machinery) and we have to decide on the product mix to maximize the profit. Certainly, we are not free to produce any product at any amount, instead for some products we have certain obligations that are constraining our decisions. Consequently we consider lower and upper demand limits for each product and the capacity limits of the production facility. We assume that the profit arising from each product is linear in the number of produced units, and given the stationary case we formulate a linear program to model the optimal product mix problem. In this fist step we do not consider uncertain demand but we indicate that stochastic programming and sensitivity analyses are apparent ways to deal with random demand. For robust planning approaches we refer to (Hood, Bermon, and Barahona 2003; Barahona, Bermon, Gunluk, and Hood 2005). Here, the objective is demand satisfaction and minimization of the number of used tools. In (Habla and Mönch 2008) you can find a minimum cost formulation that considers noisy demand.

The paper is organized as follows. In Section 2 we discuss the related literature. Also, we introduce the basic concepts of load balancing, connected components and resource pools. In Section 3 we define a (global) linear program for the optimal product mix problem. In addition, we discuss an extension of this model. In Section 4 we propose a decomposition approach for the (global) product mix problem by iteratively solving small-sized linear sub problems. We evaluate both methods considering a set of randomly generated instances. For this, we describe a benchmark scheme in Section 5. Finally, we present the results of our computational experiments in Section 6.

2 LOAD BALANCING, CONNECTED COMPONENTS AND RESSOURCE POOLS

In this section we will briefly introduce the notions of the static capacity problem and the concept of load balancing and resource pooling. For more details on capacity planning in semiconductor industry we refer to (Bermon and Hood 1999; Geng and Jiang 2009). For the theoretic concept of resource pooling in high traffic we refer to (Gold 2004; Harrison and Lopez 1999; Harrison and Williams 2007). We begin with an explanation of Figure 1 which displays a simple linear capacity planning problem. The static planning process starts with the demand of three products which are identified by the routes $route_I$, $route_{II}$ and $route_{III}$. The primary demand or the arrival rate is a input of the problem, in this example the rate is given in lot starts per week. Each route consists of a sequence of processing steps or simply jobs. Some of these jobs are very similar because they have the same pattern of capacity consumption regarding the resources (e.g. tools). For instance it is not always necessary to reflect all different products and jobs, because some routes will have the same (or very similar) specifications for some of the processes. So, by combining these jobs into so called job classes we are downsizing the capacity planning problem. Each job class j has a demand λ_j which needs to be satisfied by allocating the demand to the resources or service stations ($i \in I$). In the example we can see, that the going rate λ_j has the same dimension like the primary demand. To satisfy the demand we need to distribute the demand λ_j to the resources in a feasible way. I.e. the variable x_{ji} holds the number of units of job class j that are allocated to resource i. Each 'activity' x_{ji} has a corresponding service rate of μ_{ji} units per time unit (e.g. lots per hour). The depicted solution x is feasible but it doesn't seem to be a very good choice. For example, if the objective is to minimize the maximum load we can observe that we can get a more balanced solution (compared to the depicted one) if we shift some lots from resource a to resource b.
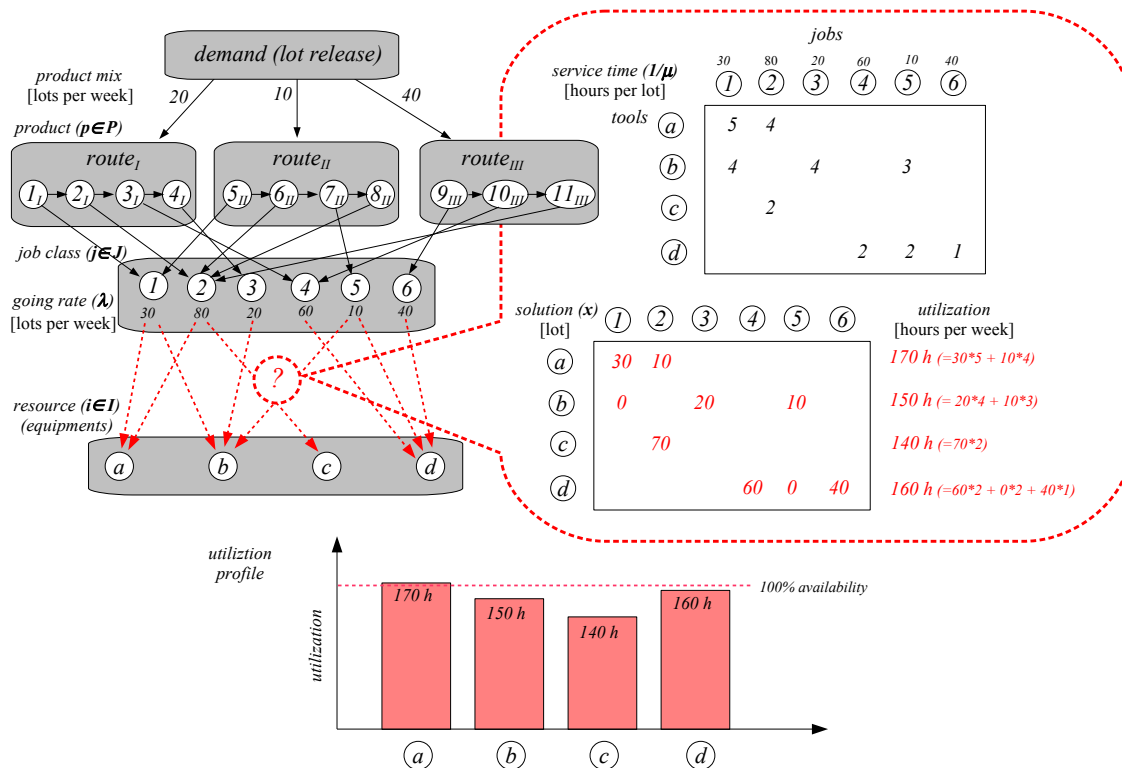

Figure 1: Capacity planning – given a certain demand of lots we have to assign the resulting job classes to the resources which thereby results in a corresponding utilization profile.

If we want to avoid bottlenecks we may minimize the maximum load as an intuitive objective. We will now formalize this problem in this way:

$$\min \rho \tag{1}$$

$$s.t. \quad \sum_{i \in I} x_{ji} = \lambda_j \qquad (\forall j \in J), \tag{2}$$

$$\sum_{j \in J \wedge \mu_{ji} > 0} \frac{x_{ji}}{\mu_{ji}} \le \rho \qquad (\forall i \in I), \tag{3}$$

$$x \ge 0. \tag{4}$$

If x_{ji} is the number of units of job class j assigned to resource i, then the set of constraints (2) guarantees the flow balance. Since μ_{ji} is the given service rate we also know that $\frac{1}{\mu_{ji}}$ is the time needed to service one unit from job class j on resource i. If $\mu_{ji} = 0$ then this activity is not part of the model. The constraints (3) are therefore limiting the capacity with ρ as an upper bound. The variable ρ is minimized and in total, the maximum load is minimized. By transforming the variables $\tilde{x}_{ji} \leftarrow \frac{x_{ji}}{\mu_{ji}}$ (which can be interpreted as the time invested in the activity x_{ji}) we get the following equivalent linear program, that highlights the queuing aspect:

$$\min \rho \tag{5}$$

$$s.t. \quad \sum_{i \in I} \tilde{x}_{ji} \mu_{ji} = \lambda_j \qquad (\forall j \in J), \tag{6}$$

$$\sum_{j \in J} \tilde{x}_{ji} \le \rho \qquad (\forall i \in I), \tag{7}$$

$$\tilde{x} \ge 0. \tag{8}$$

Solving the example in Figure 1 with a simplex-based method we get the following solution $x = (20, 0, 10, 20, 10, 80, 60, 0, 40)$ with respect to the following activities $\{(1,a), (2,a), (1,b), (3,b), (5,b), (2,c), (4,d), (5,d), (6,d)\}$. We observe that the utilization profile is $u = (100, 150, 160, 160)$. We have to emphasize that this kind of objective does not cover all practical needs and therefore we can find several kinds and combinations of objectives in the literature. One of these considerations is that we may also be interested in minimizing the number of resources with maximum load. One approach to tackle this problem is based on convex quadratic programming (Gold, H. 2008). The concept also divides the resources into disjoint 'resource pools' with a homogeneous load distribution that minimizes the maximum load on each level in a hierarchical cascaded way. Using this method, we find the following solution x and the corresponding utilization profile u: $x = (23, 7.5, 7, 20, 10, 72.5, 60, 0, 40)$, $u = (145, 138, 145, 160)$. We can see that for this solution there is only one resource (d) with maximum load, while the resources $\{a, c\}$ are balanced and the resource $\{b\}$ is on the lowest level. Therefore we get three resource pools. To introduce the concept of connected components we define that two resources are connected if there exists a common job class, which they are both able to service. Therefore a connected component contains resources that are directly or indirectly linked by job classes. We will synonymously use the the term 'closed machine set' (or *cms*) for connected components. In practice it is very unlikely that all resources are connected therefore it is possible to decompose the problem in disjoint problems. In case that the objective is also separable we can considerably reduce the complexity of the problem.

3 PROBLEM STATEMENT AND MODEL

In Section 2 we considered the problem of distributing the load optimally while the primary demand (product mix) was a input parameter. In this section the quantity of the products and the load distributions are part of the decision variables and we will formulate the corresponding product mix problem, and finally we will propose some possible extension of the model. First we introduce some indices, variables and parameters:

- indices
    - product index p,
    - job class index j,
    - resource index i,
    - activity index k (a feasible combination of job class j and resource i),
- variables
    - product mix $y = (y_p)$,
    - number of steps for an activity $x = (x_k)$,
- input parameters
    - technology matrix $D = (d_{jp})$,
    - capability matrix $A = (a_{jk})$,
    - service time matrix $R = (r_{ik})$, for $\mu_k > 0$ we set $r_{ik} = \frac{1}{\mu_k}$.
    - profit $c = (c_p)$,
    - demand limits Δ^- and Δ^+
    - maximum capacity $\rho_{max} = (\rho_{max_i})$.

In analogy to the formulation in (Hager and Spannraft 2009) we present a simple version of the optimal product mix problem:

$$\max c^T y \tag{9}$$

$$Ax = Dy, \tag{10}$$

$$Rx \leq \rho_{max}, \tag{11}$$

$$x \geq 0, \tag{12}$$

$$\Delta^- \leq y \leq \Delta^+. \tag{13}$$

For modeling additional capacity (e.g. investment or silicon foundries) and the aspect of soft bottlenecks we propose to add the following variables and parameters to the problem (9)-(13):

- additional variables
    - ρ_1^+ the consumed amount of the bottleneck capacity on equipment i (makes i a soft bottleneck),
    - ρ_2^+ the amount of additional capacity of equipment i (invest and out-sourcing),
- additional input parameters
    - $g = (g_i)$ cost for one additional capacity unit,
    - $h = (h_i)$ bottleneck budget consumption for one additional unit of resource i,
    - I_1 available bottleneck budget,
    - I_2 available budget for additional capacity,
    - $R_1^+ = (R_{1_i}^+)$ maximum available bottleneck capacity of type i (soft bottleneck capacity),
    - $R_2^+ = (R_{2_i}^+)$ maximum available additional capacity of type i.

This leads us to the following (extended) model:

$$\max c^T y + g^T \rho_2^+ \tag{14}$$

$$Ax = Dy, \tag{15}$$

$$Rx \leq \rho_{max} + \rho_1^+ + \rho_2^+, \tag{16}$$

$$h^T \rho_1^+ \leq I_1, \tag{17}$$

$$g^T \rho_2^+ \leq I_2, \tag{18}$$

$$\rho_1^+ \leq R_1^+, \tag{19}$$

$$\rho_2^+ \leq R_2^+, \tag{20}$$

$$x, y, \rho_1^+, \rho_2^+ \geq 0, \tag{21}$$

$$\Delta^- \leq y \leq \Delta^+. \tag{22}$$

Remark 1 In practice we may choose the vector ρ_{max} in such a way that if the corresponding limit is reached we still won't expect to see cycle time deviations on the corresponding tools. For allowing 'soft' bottlenecks we have to choose h, R_1^+ accordingly. E.g. by setting $R_1^+ = 0.05\rho_{max}$, $h^T = \frac{1}{\rho_{max}^T}$ and $I^+ = N$ we allow to add 5% to the maximum capacity using the bottleneck capacity and in total not more than the equivalent of N resource units. In this example we suppose that a 5% increase will not result in a cycle time deviation if appropriate bottleneck management measures are taken into account. We also remark that in this formulation we have a limit on the total bottleneck excess of all resources, but the number of resources that show an excess are not limited by a constraint (continuous formulation). Obviously, the parameters g, R_2^+ can be used to model the effect of additional capacity.

Remark 2 In analogy to (14)-(22) several other extensions of model (9)-(13) are possible, too. E.g. number and location of allowed bottlenecks, 'cold steel' (machine shutdowns) and sensitivity analyses. Most of this extension lead to Integer Programming formulations which we will not discus in this paper.

To estimate the size of the linear program (9)-(13) we notice that the cardinality of the set of activities K is equal to the number of nonzero entries in the service time matrix and let L be the average number of job classes for each product. We summarize that the number of rows, columns and nonzeros of (9)-(13) can be estimated according to the following formulas:

- number of rows: $|J| + |I|$,
- number of columns: $|P| + |K| + 1$,
- number of nonzeros: $|P| \times L + 2|K|$.

We notice that for this formulation it is not possible to directly take advantage of a decomposition into connected components, because it is very likely that the products are connecting all resources and therefore the size of this problem will be quite large. For example if we consider 1000 equipments, 1000 products and 5000 job classes with average of 5 released equipments we will get $|K| = 25000$ and if each product has 100 job classes ($L = 100$) in average we get in total: 6000 rows, 26001 columns and 152000 nonzeros. In Section 4 we propose an algorithm that separates the problem into a product mix step and a resource allocation step that are iteratively solved. The product mix step does not involve job classes and the resource allocation step is a decomposable capacity planning problem. Due to the complexity reduction and the opportunity to employ parallel computations we expect to gain a considerable speed up for larger instances.

4 A DECOMPOSTION HEURISTIC

To solve the program defined in (9)-(13) we are proposing a decomposition heuristic that is based on connected components and resource pooling. We introduce the master problem:

$$\max \quad c^T y \tag{23}$$

$$Ty \le \rho_{max}, \tag{24}$$

$$0 \le \Delta^- \le y \le \Delta^+. \tag{25}$$

The resource consumption matrix T contains factors that explain the resulting utilization profile for a given product mix. I.e. T_{ip} is the resource consumption of one unit of product p on equipment i. We can think of these factors as an estimation that is recursively improved by solving static capacity problems that tend to the best product mix. To formulate the algorithm we abbreviate the master problem as $(y,z) \leftarrow MP(c,T,\Delta^-,\Delta^+)$ and for the static capacity problem we use the following notation $(x,\rho) \leftarrow LB(R,A,\lambda)$. The function LB is minimizing the maximum load using the resource pooling concept on all levels. In case that we have more than one connected component we are also able to decompose and parallelize the calculation. For the sub problem we take the output of the master problem and we apply $LB(R,A,Dy)$. Part of the output of the algorithm is given by the solution x which is used to update the load factors. We assume that Δ^- is feasible, i.e. the incurring load resulting from the arrival rate $\lambda = D\Delta^-$ does not exceed the load limit ρ_{max}. This procedure is explained in Algorithm 1. The input parameter ε is the tolerance limit for recognizing improvements in the objective.

Algorithm 1 product mix optimization (simple decomposition)

---

input: $D,A,R,c,\rho_{max},\Delta^-,\Delta^+,\varepsilon$
output: y
$y \leftarrow 1$, $\lambda \leftarrow Dy$
$(x,\rho) \leftarrow LB(R,A,\lambda)$
$\forall_p \forall_i : \quad T_{ip} \leftarrow \sum_{j:d_{jp}>0} x_{ji} \mu_{ji} \frac{d_{jp}}{\lambda_j}$
$y \leftarrow \Delta^-$, $\lambda \leftarrow Dy$, $z_0 \leftarrow c^T y$
$(x,\rho) \leftarrow LB(R,A,\lambda)$
$\forall_{p:y_p>0} \forall_i : \quad T_{ip}(x) \leftarrow \sum_{j:d_{jp}>0} x_{ji} \mu_{ji} \frac{d_{jp}}{\lambda_j}$
$i \leftarrow 1$
while $(i = 1) \vee (z_i - z_{i-1} > \varepsilon)$ **do**
  $(y,z_i) \leftarrow MP(T,c,\rho_{max},\Delta^-,\Delta^+)$
  $\lambda \leftarrow Dy$
  $(x,\rho) \leftarrow LB(R,A,\lambda)$
  $\forall_{p:y_p>0} \forall_i : \quad T_{ip}(x) \leftarrow \sum_{j:d_{jp}>0} x_{ji} \mu_{ji} \frac{d_{jp}}{\lambda_j}$
  $i \leftarrow i+1$
end while

Theorem 1 Algorithm 1 terminates with a feasible solution.

Proof.   Each generated solution y is feasible (including the first one), because the load factors in T represent a feasible marginal load distribution. The solution of the load balancing problem aims to improve the capacity consumption on the resources with a high load without generating infeasibilities and therefore y is still feasible for the master problem with the updated load factors, hence the objective function is non-decreasing. We assume that there are no products that have no capacity consumption, so we expect that the profit is bounded and we get a convergent sequence of the objective value. Therefore Algorithm 1 will terminate with a feasible solution. □

5 DESIGN OF EXPERIMENT

We are investigating the effect of different factors of the instances on the potential improvement of the profit and the quality of the heuristic. For this reason we implemented a program to produce benchmark instances with different size and structure by varying the control parameters (you can find the octave code and the instances at the following URL: http://homepage.univie.ac.at/martin.romauch/goldenmix/). Some of the control parameters of the generator are directly effecting the factors (like the number of products), but others (e.g. the density of the service time matrix) are dependent on several control parameters. We generate an initial product mix y_0 and we define the limits for y to be 5% below and 5% above y_0, i.e. $\Delta^- = 0.95 y_0$, $\Delta^+ = 1.05 y_0$ and ρ_{max} is the minimum ρ for y_0. The solution y_0 is constructed in such a way that there is at least on resource in each component that is loaded close to ρ_{max}. To produce a structural broad set of instances of different size we consider a factorial design based the settings of the control parameters in Table 1.

Table 1: The parameters in this table are used to make the factorial design of the experiment

control parameter	settings	number of settings
cms size	small, medium, large	3
number of cms	small, medium, large	3
variance parameter for R	small, medium, high	3
density parameter for A	sparse, medium, dense	3
number of products	small, medium, large	3

For each parameter setting we use five different seeds. Therefore the total number of instances is $3^5 \cdot 5 = 1215$. To formulate the factors and variates we refer to the initial profit z_0, the profit of the global optimum z_{opt} and the profit of the decomposition method z_{decomp} (according to Algorithm 1). Table 2 summarize the factors and variates used in the analysis. To standardize the variation we sometimes use a statistic for the coefficient of variation $CV(X) = \sqrt{\mathbb{V}\mathrm{ar}(X)}/\mathbb{E}(X)$.

Table 2: The factors and variates in this table are used for the analysis. Since the problem is decomposed into sub components the letter τ indicates the corresponding *cms*. E.g. A_τ refers to the capability matrix of component τ.

factor or variate	formula or symbol	values or range
number of cms	Q	$\{20, 40, 60\}$
number of resources	$neq = \|I\|$	$[57; 2349]$
number of job classes	$njc = \|J\|$	$[74; 1500]$
number of products	$nprod = \|c\|$	$[5; 1132]$
average density of $A_\tau = (a_{jk}^{(\tau)})$	$density = \mathrm{avg}_{\tau=1}^{Q} \left(\dfrac{\|\{k : a_{jk}^{(\tau)} > 0\}\|}{\|I_\tau\|\|J_\tau\|} \right)$	0-50%
relative variance of service time	$rttvar = \mathrm{avg}_j(CV(R_j\|_{>0}))$	0-35%
coefficient of variation of profit	$CV(c)$	11-36%
maximum improvement	$potential = \dfrac{z_{opt} - z_0}{z_0}$	0-4.75%
gap	$gap = \dfrac{z_{decomp} - z_0}{z_{opt} - z_0}$	0-100%

6 RESULTS

Based on the experiments on the 1215 instances from section 5 we are presenting our main results about the influence of the structure of the problem (parameters that describe the instance) on the profit increase and on the solution quality of the heuristic method. I.e. we are measuring the effect of changing the product mix in a 5 % corridor on the relative profit improvement – and for reporting the quality of the heuristic we investigate how much of the maximum profit increase is covered by the decomposition method. We will start with explaining the Figures 2 to 5 to summarize the main results of the experiment. Finally we will also showcase a short study with real word data about these effects and sensitivity issues.
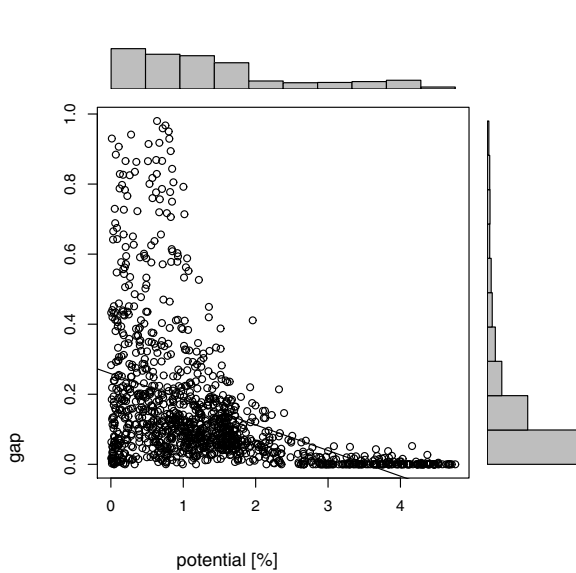
Figure 2: The relative potential (in %) and the relative gap *gap* have a negative correlation.
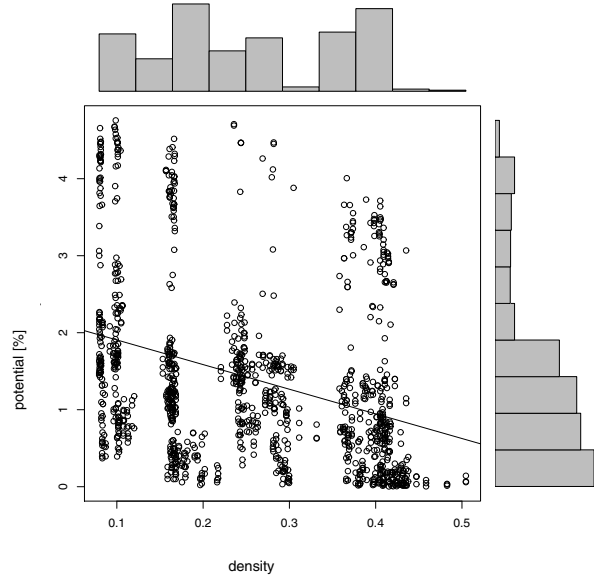

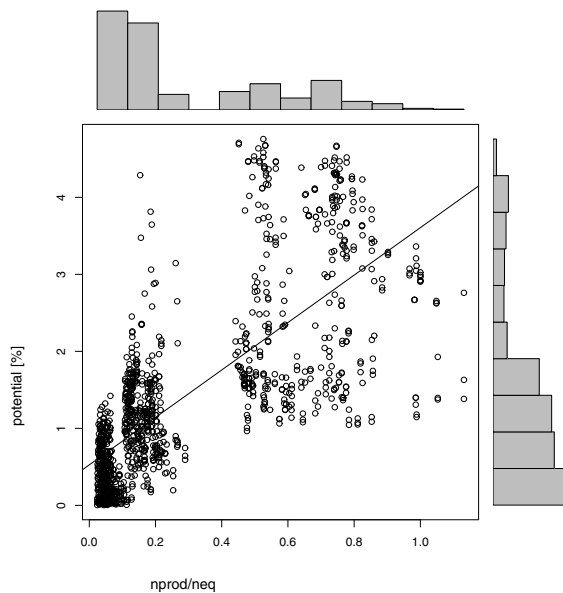Figure 3: The potential (in %) and the density have a negative correlation

Figure 4: The fraction $\frac{nprod}{neq}$ and the relative potential are positively correlated.
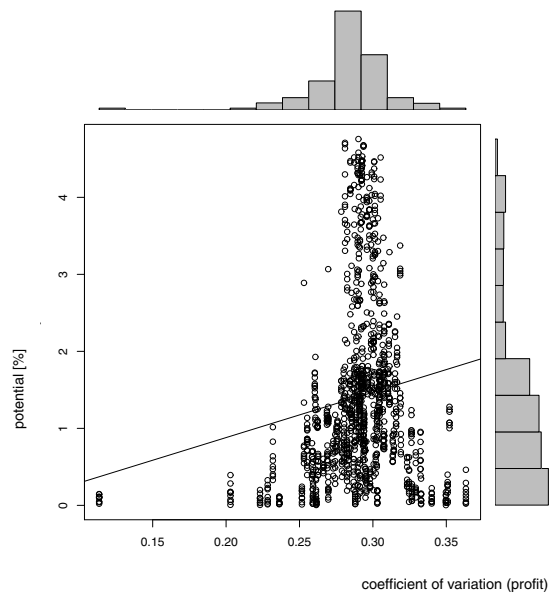

Figure 5: The potential (in %) and the coefficient of variation of the profit have a positive correlation

For computing the optimal solution of the product mix problem (global approach) the commercial solver IBM ILOG CPLEX is used on a quad core (2.6GHz) computer with 4 GB RAM. For solving the LP (sub) problems of the decomposition approach we use the non-commercial solvers GLPK (Makhorin, A. 2012) and LinProg on the same PC; in some cases (especially for large problem instances) they were not able to find the optimal solution of the global problem. Now we are interested in the performance of the decomposition approach. Therefore, we display the correlation between the relative potential and the relative gap (cf. Table 2) in Figure 2. Here we can see that – with an increasing optimization potential – the decomposition heuristic gradually approaches to the global optimum. Now we are interested in the dependence of optimization potential to the parameters varied in the benchmark scheme. For this, its correlation to the density of the dedication matrix as well as the correlation to the fraction *nprod/neq* is visualized in Figure 3 and 4. As we can see, especially in cases with a high number of products and complex dedication constraints (low density in dedication matrix) a high optimization potential exists. Referring to the generation of benchmark instances, the entries of the profit vector *c* are modeled as independent and uniformly distributed random variables. According to the central limit theorem, the sum of squares tends to a normal distribution, and therefore we observe (cf. Figure 5) that the profit's marginal distribution of the coefficient of variance is approximately bell shaped. According to the scatter plot of the Figure 5 we can also see that the optimization potential is following this tendency, which is not obvious and at the moment we cannot provide a satisfactory explanation.

Overall we have seen that a profit improvement of approximately 3% is possible even if each component in the initial solution is loaded close to ρ_{max}. If the product mix portfolio is adaptable in the range of +/- 5% this also matches with our practical experiences in real world application. Here, next to the maximization of profit, this model – as well as some extensons of them (cf. remark 2) – helps us to identify products which are drivers for bottlenecks and to estimate how the tools will react on product changes (sensitivity analysis). Because we have a direct interaction between a given product mix and the resulting equipment utilization we can fast and target oriented decide how to react on changing demands. Thereby, the decomposition on several stages (job class concept, connected components, optimization) allows a calculation on the highest granularity level, that means on the basis of all routes, all products, all operations, all equipments and all dedications in the MES system with an overall calculation time of only a couple of minutes.
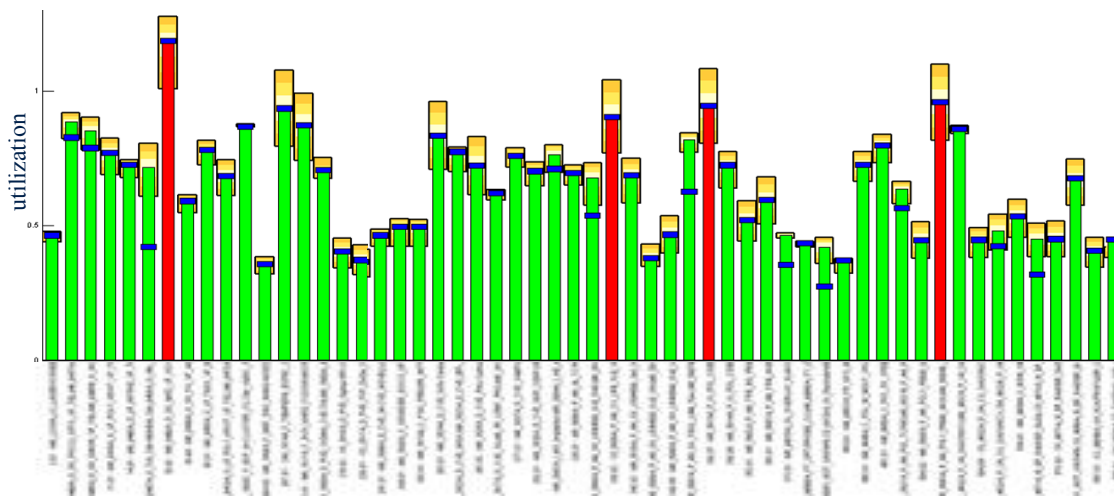

Figure 6: Sensitivity analysis – effects of product mix fluctuations (current mix).

Figure 6 and 7 show the solution of a product mix optimization on the basis of an altered fab-data set. On the x-axis the connected components (work centers) are displayed. The length of the bars represent the maximum tool utilization within the work center resulting from the current product mix. The average

utilization is marked blue, different sensitivity intervals are displayed yellow (cf. Figure 6) and red bars show bottleneck work centers.

Now, the effect of an product mix adaption is shown in Figure 7. It leads to a significant load reduction (black bars) on the bottleneck work centers. Thereby the overall fab load (layer starts per week) is not changed – but the product mix.
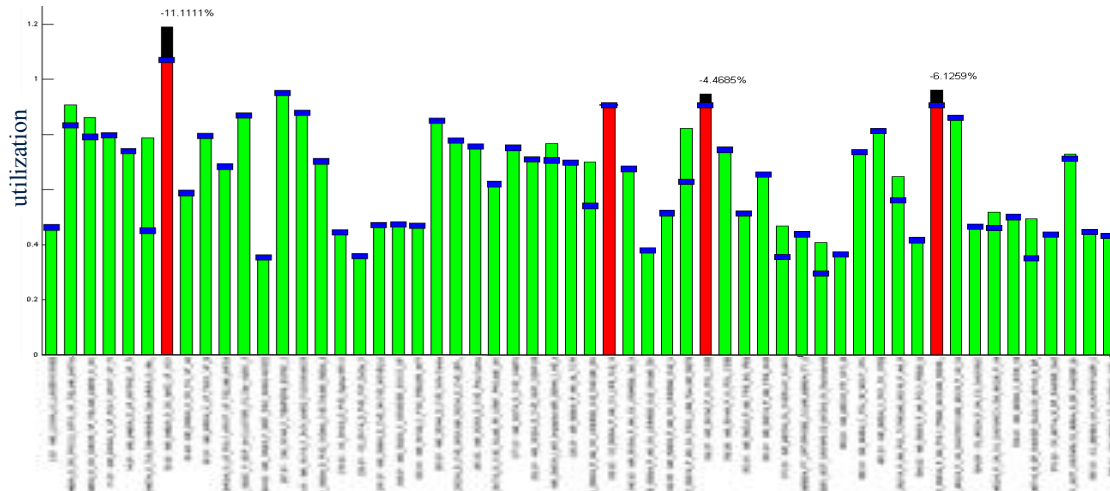

Figure 7: Optimized product mix – product mix change relaxes the bottleneck situation .

7 CONCLUSIONS AND FURTHER RESEARCH

The contribution of this paper can be seen as twofold. On the one hand the theoretical part: We have investigated which structural parameters affect the maximum profit improvement of the product mix problem. Here we could show, that the number of products and the variation of the profit are affecting the relative potential in a significant way. Furthermore, we have introduced a decomposition heuristic to the product mix problem and we have inspected them on the basis of randomly generated test instances. Here, we were able to proof that this method converges to good quality solutions within reasonable time. Due to its reduction of complexity it is also interesting that we could show that within this concept already free solvers like GLPK are sufficient to solve large instances. For some instances with a small optimization potential we experienced that the heuristic method got stuck in local optima and we are currently working on a promising improvement of the method.

On the other hand the practical part: We drafted, that the approach covers numerous problems within semiconductor industry. The method helps us in evaluating risks, in the improvement of lot release strategies, for bottleneck management issues and for sensitivity analysis's. Furthermore, we are able to make capacity investment decisions more precise and faster.

In future we will focus our work on optimizing product mix and profit for the whole supply chain within a semiconductor company. In combination with dynamic techniques this method will support an optimized product flow and reduce logistics costs within the supply chain.

ACKNOWLEDGMENTS

REFERENCES

Barahona, F., S. Bermon, O. Gunluk, and S. Hood. 2005. "Robust Capacity Planning in Semiconductor Manufacturing". *Naval Research Logistics* 52:459–468.

Bermon, S., and S. Hood. 1999. "Capacity optimization planning system (CAPS)". *Interfaces* 29 (5):31–50.

Geng, N., and Z. Jiang. 2009. "A review on strategic capacity planning for the semiconductor manufacturing industry". *International Journal of Production Research* 47 (13):3639–3655.

Gold, H. 2004. "Dynamic Optimization of routing in a Semiconductor Manufacturing Plant". In *Operations research proceedings 2004: selected papers of the annual international conference of the GOR; jointly organized with the NGB*, edited by P. K. Hein Fleuren, Dick den Hertog, 76–83. Tilburg, The Netherlands: GOR and NGB.

Gold, H. 2008. "Lastverbund, european patent EP1567921". July 2012. https://data.epo.org/publication-server/rest/v1.0/publication-dates/20081112/patents/EP1567921NWB1/document.html.

Habla, C., and L. Mönch. 2008, December. "Solving volume and capacity planning problems in semiconductor manufaturing: A computational study". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 2260–2266. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hager, V., and K. Spannraft. 2009. *A product mix and a material flow problem concerning the semiconductor manufacturing industry*. Master thesis, Department of Business Administration, University of Vienna, Vienna, Austria. Available via http://othes.univie.ac.at/8092/ [accessed May 30, 2012].

Harrison, J. M., and M. J. Lopez. 1999. "Heavy traffic resource pooling in parallel-server systems". *Queueing Systems* 33:339368.

Harrison, J. M., and R. J. Williams. 2007. "Workload Interpretation for Brownian Models of Stochastic Processing Networks". *Mathematics of Operations Research* 32 (4):808–820.

Hood, S. J., S. Bermon, and F. Barahona. 2003. "Capacity Planning Under Demand Uncertainty for Semiconductor Manufacturing". *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING* 16 (2): 273–280.

Makhorin, A. 2012. "GNU Linear Programming Kit (Version 4.34)". http://www.gnu.org/software/glpk/glpk.html.

AUTHOR BIOGRAPHIES

ANDREAS KLEMMT received his master's degree in mathematics in 2005 and doctor's degree in electrical engineering in 2011 from the Dresden University of Technology. He works as operations research and engineering expert at Infineon Dresden. His current research interests are mathematical programming, capacity planning, production control, and simulation and optimization. His email is Andreas.Klemmt@infineon.com.

MARTIN ROMAUCH is assistant professor at the University of Vienna and works in the Production and Logistics division of the Department of Business Administration. His research interests include facility location, mathematical modeling, metaheuristics, supply chain management and real world applications. He received his master's degree in applied mathematics at the University of Klagenfurt and a doctor's degree from the University of Vienna. Besides his academic career he gained practical experience working in the operations research & engineering division of Infineon Technologies for a couple of years. His email is Martin.Romauch@univie.ac.at.

WALTER LAURE has a diploma in Technical Mathematics from the Technical University of Graz (Austria). He joined Infineon Technologies in 1996 and was responsible for specification and implementation of several software projects. After that he build up a simulation group to apply simulation methods to efficiency improvement projects. His main experience covers beside simulation detailed aspects of mathematical

optimization, data modeling and statistics. This methods are used to optimize shop floor control, tool capacity planning and supply chain planning. His email is Walter.Laure@infineon.com.