Discrete Optimisation and Real World Problems

Josef Kallrath and Anna Schreieck

BASF-AG, ZX/ZC-C13, D-67056 Ludwigshafen

Abstract. In the global economy, proper organisation and planning of production and storage locations, transportation and scheduling are vital to retain the competitive edge of companies. The planning and scheduling problems involved are immensely complex. Computer-based optimisation techniques are the best means of obtaining viable solutions, but until now the mixed integer programs developed have been able to deal only with simple problems. The more important larger problems have generally been solved using ad-hoc heuristics which often produce incomplete and less satisfactory solutions. Today, the development of new algorithms, software and hardware is leading to the provision of mathematical applications and tools which allow the solution of these larger problems in acceptable times. In this contribution two groups are addressed: on the one hand managers, and on the other hand a more technical oriented audience. The focus towards the first group is to create some attention with respect to the potential benefits of the method, to transmit a sense of what kind of problems can be tackled, and to increase the acceptance of the approach based on mixed-integer optimisation. The second group will be informed about the state-of-the-art, especially with respect to the use of high-performance computers and modern algorithmic aspects.

1 Introduction

Optimisation problems arise in almost all branches of industry or society, *e.g.* in product and process design, production, logistics, traffic control and even strategic planning. In an optimisation problem (OP), one tries to minimise or maximise a global characteristic of a process such as elapsed time or cost, by an appropriate choice of parameters which can be controlled, and under a set of constraints, linked for example to physical limits. A traditional way to develop answers to optimisation problems is to propose a number of choices for the controlled parameters, using heuristic methods. The processes under investigation are then simulated under these various options, and the results are compared. Engineers in charge of these OPs have developed intuition and heuristics to select appropriate conditions, and simulation software exists to perform the evaluation of their performance. The "traditional" techniques may lead to proper results, but there is no guarantee that the optimal solution or even a solution close to the optimum is found. This is especially troublesome for complex problems, or those which require decisions with large financial impact.

In contrast to simulation, optimisation methods search directly for an optimal solution that fulfills all restrictions and relations which are relevant for the real-world problem. By using mathematical optimisation it becomes possible to control and adjust complex systems even when they are difficult for a human being to grasp. Therefore, optimisation techniques allow a fuller exploitation of the advantages inherent to complex systems.

Classical optimisation theory (calculus, variational calculus, optimal control) treats those cases in which the parameters can be changed continuously, e.g. the temperature in a chemical reactor. On the other hand, mixed integer, combinatorial or discrete optimisation addresses parameters which are limited to integer values, for example counts (numbers of containers, ships), decisions (yes-no), or logical relations (if product A is produced then product B also needs to be produced). This discipline, years ago only a marginal discipline within mathematical optimisation, becomes more and more important.

2 A Survey of Real World Problems

The brief survey of real world problems given in this section is typical for the chemical industry but most of the topics also occur in other areas:

- blending problems (production & logistics)
- production planning (production, logistics, marketing)
- scheduling problems (production)
- process design (process industry)
- depot selection problems (strategic planning)
- network design (planning, strategic planning)

Typical for the chemical industry, but in modified form also for the mineral oil or food industry, are **blending problems**. They occur in a wide variety. In [7] a model is described for finding cost minimal blending which simultaneously include container handling conditions and other logistic constraints. Specially, companies which are in a situation to utilise the advantages of a complex production network, often with the background of several sites [7], may greatly benefit from **production planning** and **production scheduling**. Of course, scheduling problems occur also in other branches of industry. They are operational and yield detailed answers to the questions: when is the production of a specific product on a specific machine to be started ? What does the daily production sheet of a worker look like ? Scheduling problems belong to a class of the most difficult problems in discrete optimisation. Typical special structures, which can be tackled by discrete optimisation, are minimal production rates, minimal utilisation rates, minimal transport amounts: these structures lead to so-called semi-continuous variables. The question, how a telecommunication network should be structured and designed when the annual demand is known, or the question, what the traffic infrastructure should look like for a given traffic demand lead to **network design problems**. While the problems listed above can be solved with linear mixed-integer methods, problems occurring in process industry very often lead to **nonlinear discrete problems**.

3 Some Mathematical Background on Mixed-Integer Optimisation

This section provides some of the mathematical and algorithmic background. It is addressed to a more technical audience, and might be skipped by readers more interested in applications.

3.1 Linear Programming

A linear programming problem or LP in standard form is defined by

$$\begin{array}{llll} \textbf{LP:} & \text{Minimize:} & z(\mathbf{x}) = \mathbf{z}^T\mathbf{x}, & \mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n \\ & \text{Subject to:} & A\mathbf{x} = \mathbf{b} & A \in \mathcal{M}(m \times n, \mathbb{R}) \text{ matrix}, \mathbf{b} \in \mathbb{R}^m \\ & & x \geq 0 & \end{array} \tag{1}$$

Other formulations of LPs using inequalities or unconstrained variables can be mapped to this standard form. LP is mentioned in this place since very often LP-algorithms are used as a subroutine in integer programming algorithms to obtain lower bounds on the value of the integer program. One of the best known algorithms for solving LPs is the *simplex algorithm* of G.B. Dantzig (e.g. [14]) which can be understood both geometrically and algebraically. The algebraic platform is the concept of the basis $\mathcal{B}$ of A, *i.e.* a linearly independent collection $\mathcal{B}$=$\{A_{j_1},...,A_{j_m}\}$ of columns of A. The inverse $\mathcal{B}^{-1}$ gives a basic solution $\bar{\mathbf{x}} \in \mathbb{R}^n$

$$\bar{x}_j = 0 \text{ if } A_j \notin \mathcal{B} \quad , \quad \bar{x}_{j_k} = k^{th} \text{ component of } \mathcal{B}^{-1}b, k = 1, ..., m \tag{2}$$

If $\bar{\mathbf{x}}$ is in the set of feasible points $S = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \, \mathbf{x} \geq 0\}$, then $\bar{\mathbf{x}}$ is called a *basic feasible solution*. If (i) the matrix A has m linearly independent columns A_j, *i.e.* A is of rank m, (ii) the set S is not empty and (iii) the set $\{\mathbf{z}^T\mathbf{x} : \mathbf{x} \in S\}$ is bounded from above, then the set S defines a convex polytope P and each basic feasible solution corresponds to a vertex of P [13]. (ii) and (iii) ensure that the LP is neither infeasible nor unbounded, i.e. has a finite optimum. As the optimal solution of a LP is among the finite set of basic feasible solutions, the idea of the simplex algorithm is moving from vertex to vertex of this polytope to improve the objective function value. In this sense finding an optimal solution for an LP is a combinatorial problem. In each iteration, one element of the actual basis is exchanged, according to this exchange of basis variables, matrix A, and vectors $\mathbf{b}$ and $\mathbf{c}$ are transformed to matrix A', and vectors $\mathbf{b}'$ and $\mathbf{c}'$. Instead of computing these components based on the previous iteration, the revised simplex algorithm is rather based on the initial data A and $\mathbf{c}$. By this, rounding errors do not accumulate. In addition, in most practical applications A is very sparse whereas after several iterations the transformed matrix A' mostly gets much denser so that especially for large problems the revised simplex algorithm usually needs far less operations. The simplex or the revised algorithm finds an optimal solution of an LP problem after a finite number of iterations, but in the worst case the running time may grow exponentially. Nevertheless on many real world problems it performs better than polynomial time algorithms developed in the 1980's e.g. by Karmarkar and Khachian ([11], [12]).

Based on the results of Karmarkar, in the last few years a large variety of *interior point methods* has been developed (e.g. [6], [10]), so called primal-dual predictor-corrector methods already have been integrated into some LP-solvers, as XPRESS-MP. The idea of IPM's is proceeding from an initial interior point $\mathbf{x} \in S$ satisfying $\mathbf{x} > 0$, towards an optimal solution without touching the border of the feasible set S. The condition $\mathbf{x} > 0$ is guaranteed by adding a penalty term to the objective function. Thus (1) transforms to the logarithmic barrier problem (LBP):

$$\begin{array}{lll} \textbf{LBP:}\ \text{Minimize:} & z(\mathbf{x}) = \mathbf{z}^T\mathbf{x} - \mu \sum_{i=1}^{n} \ln x_i & \mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n, \mu > 0 \\ \quad\quad\ \ \text{Subject to:} & A\mathbf{x} = \mathbf{b} & A = M(m \times n, \mathbb{R}),\ \mathbf{b} \in \mathbb{R}^m \end{array} \tag{3}$$

By suitable reduction of the homotopy parameter $\mu > 0$, the weight of the penalty term successively and the sequence of points obtained by solving the perturbed problems, converges to the optimal solution of the original problem.

3.2 Mixed-Integer Optimisation

Restricting the domain of all or of a part of variables x_j of problem LP to integer values or to disjoint sets, *e.g.* $x \in [d_1, d_2] \vee x \in [d_3, d_4]$, $d_1 \leq d_2 < d_3 \leq d_4$, an integer (ILP) or a mixed-integer linear programming problem (MILP) results.

$$\begin{array}{lll} \textbf{MILP}\ \text{Minimize:} & z(\mathbf{x}, \mathbf{y}) = \mathbf{z}^T\mathbf{x} + \mathbf{h}^T\mathbf{y}, & \mathbf{x}, \mathbf{z} \in \mathbb{Z}^n,\ \mathbf{y}, \mathbf{h} \in \mathbb{R}^r \\ \quad\quad\quad \text{Subject to:} & A\mathbf{x} + B\mathbf{y} = \mathbf{b} & A \in \mathcal{M}(m \times n, \mathbb{R}) \\ & x \geq 0, \quad y \geq 0 & B \in \mathcal{M}(m \times r, \mathbb{R}),\ \mathbf{b} \in \mathbb{R}^m \end{array} \tag{4}$$

Building mixed-integer models requires great caution. Often there exist different possibilities to formulate the restrictions of an OP ([3]), sometimes adding redundant constraints makes an algorithm work faster, *e.g.* if the gap between the optimal solutions of the LP-relaxation and of the original problem is diminished by this. Even some nonlinear OPs can be transformed to MILP's using special types of discrete variables, see e.g. [15], [5].

- Logical conditions, such as "and", "or", "not", "implies", and also disjunctive constraints are formulated with *binary variables* $\delta \in \{0, 1\}$.
- Binary variables can indicate the state of a continuous variable and at the same time impose upper and lower bounds (L and U) on this variable. The constraints $x = 0 \vee L \leq x \leq U$ defining a *semi-continuous variables* x are equivalent to $L \cdot \delta \leq x \leq U \cdot \delta$, where δ is a binary variable. Some software packages offer *semi-continuous variables* to formulate this constraint directly without utilizing an additional binary variable ([1], [16]) which provides great advantages for the B&B procedure.
- *Special ordered sets of type n (SOSn)* have been developed to formulate common types of restrictions in mathematical programming. In SOS1 sets of variables exactly one variable (continuous or integer) must be non-zero.

In an SOS2 set two variables which are adjacent in the ordering of the set or one single variable must be non-zero. SOS2 sets often are used to model piecewise linear functions, e.g. linear approximations of nonlinear functions.

– Programs with products of k binary variables $\delta_p = \prod_{i=1}^{k} \delta_i,\ \delta_i \in \{0,1\}$ can be transformed directly into integer models according to

$$\delta_p \leq \delta_j,\ j = 1, ..., k \quad ; \quad \sum_{j=1}^{k} \delta_j - \delta_p \leq k - 1 \quad ; \quad \delta_j \in \{0,1\} \quad . \tag{5}$$

An overview on methods specially designed for solving mixed integer nonlinear problems is given in [8]. A great variety of algorithms to solve mixed integer OPs has arisen during the last decades. Among the best known *exact algorithms* for solving ILP's are the following methods:

enumerative methods - cutting-plane algorithms - dynamic programming

Efficient **enumerative algorithms** include pruning criteria so that not all feasible solutions have to be tested for finding the optimal solution and for proving optimality. The widely used B&B algorithm with LP-relaxation is the most important representative of enumerative algorithms and therefore discussed in more detail in the next subsection.
Cutting plane algorithms for MILP's are derived from the simplex algorithm. After computing the continuous optimum by LP-relaxation of the integrality constraints step by step new constraints are added to the MILP. With the help of these additional inequalities non integer variables of the continuous solutions are forced to take integer values, see *e.g.* [4], [12]. Cutting plane methods are not restricted to MILP's, they are used e.g. in nonlinear and nondifferentiable optimisation as well (Lemaréchal in [11]).
Dynamic programming ([12], [14]) is not a general-purpose algorithm like methods belonging to the first two groups. Originally, it was developed for the optimisation of sequential decision processes. This technique for multistage problem solving may be applied to linear and nonlinear OPs which can be described as a nested family of subproblems. The original problem is solved recursively from the solutions of the subproblems.

Furthermore there exist *heuristics*, local and global search algorithms as for instance *simulated annealing*.

3.2.1 Branch-and-Bound Algorithm The first B&B algorithm was developed in 1960 by Land and Doig. The *branch* in B&B hints at the partitioning process used to prove optimality of a solution. Lower *bounds* are used during this process to avoid an exhaustive search in the solution space. The B&B idea or *implicit enumeration* characterizes a wide class of algorithms which can be applied to discrete OPs in general.

A B&B algorithm of Dakin [4] with linear programming relaxations uses three *pruning criteria*: infeasibility, optimality and value dominance relation.
The branching in this algorithm is done by variable dichotomy: for a fractional $x_{i_0}^*$ two son nodes are created with the additional constraint $x_{i_0} \leq [x_{i_0}^*]$ resp.

$x_{i_0} \geq [x^*_{i_0}] + 1$. Other possibilities for dividing the search space are *e.g.* generalized upper bound dichotomy or enumeration of all possible values, if the domain of a variable is finite ([4], [12]). The advantage of variable dichotomy is that only simple constraints on lower and upper bounds are added to the problem.

The *search strategy* plays an important role in implicit enumeration, widely used is the depth-first plus backtracking rule as presented above. If a node is not pruned, one of its two sons is considered. If a node is pruned, the algorithm goes back to the last node with a son which has not yet been considered (backtracking). In linear programming only lower and upper bound constraints are added, the dual simplex algorithm can reoptimize the problem directly without data transfer or basis reinversion [12]. Furthermore, it is more likely that feasible solutions are found deep in the tree as experience has shown [4]. Nevertheless, in some cases the use of the opposite strategy, breadth-first search, may be advantageous.

Another important point is the *selection of the branching variable.* A common way of choosing a branching variable is by user-specified priorities, because no robust general strategy is known. Degradations or penalties may also be used to choose the branching variables, both methods estimate or calculate the increase of the objective function value if a variable is required to be integral, especially penalties are costly to compute in relation to the gained information so that they are used quite rarely [12].

The B&B algorithm terminates after a finite number of steps, if the solution space of the LP-relaxation of problem MILP is bounded.

4 A Solved Real-world Problem

An example developed in the past by BASF is a production planning system for three sites located in Germany, USA and Asia. Each of the plants can produce the same three products with equal quality in order to satisfy existing demand. The quality of products is only guaranteed if the plant operates at least on a 50% level. Otherwise there is no production. The number of change-overs per year is limited, say 5/year, to reduce risk associated with machine starting. The model describes a scenario including product change-over times dependent on production site, discrete transportation capacities, transportation times and inventory properties, and is characterized by

Plants	capacities	setup-times	utilisation rate
Inventories	capacities	additional inventory	security stock
Transport	minimal amounts	transport times	
Orders	monthly	satisfy where possible	

The **objective** is to determine production, change-overs, inventory, shipping, and sales such that demands are satisfied where possible and that the contribution margin (income minus variable cost for production, change-over, inventory, external purchase and transport) becomes maximal.

The mathematical model is described in details in [7] and leads to a mixed-integer linear programming problem with 72 binary, 248 semi-continuous and 1401 continuous variables, and eventually 976 constraints. Using XPRESS-MP on an 80386-PC it was possible to compute the first integer solution within a few minutes. The duality gap was further reduced by using appropriate cuts, which eventually allows to prove optimality within minutes. Using the model it is possible to achieve an additional profit which is of the order of a few percent of the contribution margin.

5 Parallelisation as an Answer to Increasing Complexity?

Some successfully solved real world problems in BASF demonstrate the huge potential for reducing costs, increasing efficiency and the flexible use of resources by mathematical optimisation. However, it is also observed that many problems lead to a complexity which goes beyond today's hardware and algorithmic capabilities. In some cases, it is not possible to prove optimality. To estimate the quality of the solution, save bounds are derived, instead. In order to solve complex mixed-integer models with not only a few hundred, but rather a few thousand, or even tenthousands of discrete variables, BASF initiated the project PAMIPS [9]. PAMIPS (**P**arallel **A**lgorithm and software for **M**ixed **I**nteger **P**rogramming in Industrial **S**cheduling) is a project supported under ESPRIT connecting four industrial partners and three universities. The project team tries to solve scheduling, production planning, and network design problems with parallel mixed-integer optimisation.

The exact methods briefly described in Section 3 for solving mixed-integer problems provide two different ways for the parallelisation: the combinatorial part of the algorithm and the linear program algorithm.
The combinatorial part is either a B&B or a branch-and-cut (B&C) algorithm. In both cases it is necessary to solve many LPs. Obviously, the evaluation of the subproblems may be performed by a network of parallel processors or workstations. The subproblems are more or less decoupled from each other and allow a simple parallelisation with course granularity. Positive results have been achieved [2] on a transputer system with 8 slave- and one master-processor. It was possible to get an almost-linear speed-up.

The linear optimisation kernel is much more difficult to optimise. As described in Section (3) commercial software uses two methods to solve linear programs: revised Simplex-algorithm and interior point methods. There exists attempts to parallelise the Simplex-algorithm, but they only obtained a low speed-up. Therefore, there is more optimism towards the parallelisation of interior point methods. The major numerical work of solving IPM's is to solve nonlinear systems of equations. Linearisation in combination with Newton's method leads to linear systems of equations. On that level, broad experience with parallelisation is available. The hope is to have, at the end of the project, efficient software available which has

- B&C algorithms, spezialised B&C algorithms for scheduling problems,

• parallel B&B and B&C methods,

• parallel Simplex-algorithms, parallel interior–point methods

embedded, and which allows to solve more complex problems.

6 Future Aspects: Benefits and Barriers

With the opening of markets and borders and a globalisation of world economy, problems as the one described above will increase both in number and complexity. The ability to produce complete accurate and optimal solutions to larger problems offers the potential for enormous reductions in costs, for huge increases in efficiency, and careful handling of resources. Some industry specialists estimate that savings would average 3-4% of turnover and could well be much higher. In particular, it becomes possible to exploit the inherent advantages and synergies in highly connected production networks.

In some areas solution speed is critical to the success of the method and the acceptance of technology. Scheduling problems, for instance, must be solved in minutes when sudden changes occur in a factory and personnel has to be reallocated to machines.

Technological progress is on its way, both on the algorithmic or software side and on the hardware side within the ESPRIT funded project PAMIPS. The mathematical representation (model formulation) is improved. The very latest ideas in solution algorithms (specialised B&C algorithms for scheduling problems) are incorporated. Via parallelisation many (say, a hundred) processors work cooperatively to solve big problems in a hundredth of the time one processor would need.

Unfortunately, the support of expert decision and heuristics by mathematical models and methods is still far from being widely accepted. Very often, analysts experience great reservations when talking to people working in production, logistics or marketing. There is a psychological and/or cultural barrier. Experts are used to decision taking based on experience and heuristics which are difficult to express explicitly. The approach to achieve objective solutions which can be controlled on a quantitative basis is new. It may create unconscious fears, and may in addition require a huge effort to explain the problem of interest to a non-specialist with the appropriate degree of completeness and accuracy. Indeed, on the one hand the mathematical kernel of the application operates as a black box usually difficult to understand for non-mathematicians. On the other hand, experts are afraid to lose influence and acknowledgment when outsiders, in this case mathematicians, can produce solutions which prove to be better in terms of costs, contribution margin, utilisation rate or some other valuable quantity, when compared to their solutions. At least 50% of all real-world problems by mathematical optimisation methods is related to the psychology with respect to increase acceptance, removing reservations and fears. Thus, besides technological efforts there should be a strong investment in improving the awareness and acceptance of mathematical optimisation applied to real-world problems.

Mathematical methods and techniques cannot replace human inventiveness or decisions, but they can very well provide a quantitative basis for these decisions and allow to cope most successfully with complex problems.

References

1. Ashford R.W. , Daniel R. : Some Lessons in Solving Practical Integer Problems. J. Opl. Res. Soc. **43(5)** (1992), 425–433.
2. Ashford R.W., Connard P. & Daniel R. : Experiments in Solving Mixed Integer Programming Problems on a Small Array of Transputers. J.Opl Res.Soc **43**(5), (1992), 519-531.
3. Barnhart C., Johnson E.L., Nemhauser G.L. : Formulating a Mixed-Integer Programming Problem to Improve Solvability. In: Operations Research 41 (1993) 6, pp. 1013-1019
4. Burkhard R.E.: Methoden der Ganzzahligen Optimierung. Springer, Wien, New York, 1972.
5. Garfinkel R.S., Nemhauser G.L.: Integer Programming. John Wiley & Sons New York, 1972.
6. Gonzaga C.L. : Path-Following Mtheods for Linear Programming. SIAM Review 34 (1992) 2, pp. 167-224.
7. Kallrath J. : Diskrete Optimierung in der chemischen Industrie. in: Mathematik in der Praxis - Fallstudien aus Industrie, Wirtschaft, Naturwissenschaften und Medizin. Springer Verlag, Heidelberg, 1995.
8. Leyffer S. :Deterministic Methods for Mixed Integer Nonlinear Programming. PhD Thesis, Department of Mathematics and Computer Science, University of Dundee, Dundee, 1993
9. Schreieck A., Lücking F. : PAMIPS Esprit Project 8755, Marketing Management - Public Report, BASF-AG, ZX/ZC-C13, D-67056 Ludwigshafen, 1995.
10. Lustig I.J., Marsten R.E., Shanno D.F. : On Implementing Mehrotra's Predictor-Corrector Interior-Point Method for Linear Programming. SIAM J. Optimisation 2 (1992) 3, pp.435-449.
11. Nemhauser G.L., Rinnooy Kan A.H.G., Todd M.J. (eds.): Optimisation. North Holland, Amsterdam, 1989.
12. Nemhauser G.L. , Wolsey L.A. : Integer and Combinatorial Optimisation. John Wiley & Sons, New York, 1988.
13. Papadimitriou C.H., Steiglitz K. : Combinatorial Optimisation: Algorithms and Complexity. Prentice Hall, Englewood Cliffs, New Jersey, 1982.
14. Ravindran A., Phillips D.T., Solberg J.J. : Operations Research. Principles and Practice. John Wiley & Sons, New York, 1987.
15. Williams H.P. : Model Building in Mathematical Programming. John Wiley & Sons, Chichester, New York, 1990.
16. Ashford R.W. , Daniel R. : XPRESS-MP Manual. Dash Associates, Blisworth House, Northants NN73BX, UK, 1995.

This article was processed using the LaTeX macro package with LLNCS style