Robust Discrete Optimization

Claus-Dieter Volko

July 6, 2012

1 Introduction

This paper deals with a variant of combinatorial optimization problems that considers uncertainty of some of its parameters. Combinatorial optimization problems are about selecting a subset of a set of discrete elements so that a given number of conditions are met and the solution fulfils an optimality criterion. Special variants of combinatorial optimization problems, such as linear programs, deal with systems of equations and inequalities which are to be solved in an optimal way. Sometimes it may be useful to allow the parameters of these problems to float to a certain extent since it is not always clear what exact value a factor in one of the inequalities will have; it is possible that only a range in which this factor can be is given.

Although classical combinatorial optimization problems have been thoroughly investigated and a bunch of well-known methods for solving them are known, they are still a subject of research. Even more so are problems in which only ranges for the factors are defined instead of exact values. In the course of the decades several possible approaches have been published, and the rather elegant method of Dimitris Bertsimas and his colleagues from MIT called "Robust Optimization" is still rather young.

If the parameters are integers, the related variant of this method is called "Robust Discrete Optimization". The following paper is supposed to be an easy-to-understand introduction to this topic.

2 Combinatorial optimization problems

Often combinatorial optimization problems can be represented by systems of inequalities the optimal solution to which is to be found by a computer program. There is usually a condition saying that some product has to be maximized or minimized, and the solution should be as close to the maximum or the minimum of this product as possible.

The general form of such a problem looks as follows:

$\max c'x$ where
$Ax \leq b$
$l \leq x \leq u$ (1)

In these formulas, c, x, l and u are vectors of dimension m, and A is a $m * n$ matrix. The vector c' is the transposed vector of c, which can be regarded as the cost vector. The vector x is the solution vector. The costs are defined in advance. What the program is supposed to find is a solution vector that makes the scalar product of cost vector and solution vector as great as possible, while the individual items of the solution vector have to be greater or equal to the corresponding items in the given vector l (lower bound) and lower or equal to the corresponding items in the given vector u (upper bound). The scalar product of each row of A with the vector x is computed and it is compared to the corresponding item in vector b. Only if all of these equations are fulfilled, the solution is valid.

Such combinatorial optimization problems can be automatically solved by computers using well researched methods such as linear programming or integer linear programming (in case the solution vector x may contain only integers), respectively. Solution algorithms such as the Simplex method, Benders decomposition and Dantzig-Wolfe decomposition can be found in textbooks on combinatorial optimization, e.g. (8).

3 Optimization problems with uncertainty

Uncertainty in an optimization problem can appear in the three following forms:

1. The cost vector c is not precisely defined
2. The coefficients of the matrix A are not precisely defined
3. The vector b is not precisely defined

Of course two or all three cases may appear at the same time.

Regarding the vector b, no special new algorithm is needed to solve problems involving its uncertainty, since it suffices to introduce a new variable x_{n+1} and write $Ax - bx_{n+1} \leq 0$, thus augmenting the matrix A to include b.

In general uncertainty can be expressed either as a statistical distribution or as a range. In the former case, the statistical approach that is explained in the next chapter (stochastic programming) can be applied. In the latter case, the robust approaches by Bertsimas and others come to play.

4 Statistical approach

The statistical approach (stochastic programming) was already proposed in the 1950s by George B. Dantzig. (7) This approach requires that the distribution function of the data must be known in advance. Otherwise it will not lead to an optimal result. This can be a real problem in many cases since it is often difficult to obtain a statistical model of the data. For this reason the statistical approach is not suitable if no accurate model is known, which is usually the case in assessing customer demand for a product, for example. Estimation errors have especially dire consequences in industries with long production lead times, such as the automotive, retail and high-tech industries: They result in stockpiles of unneeded inventory or lost sales and customers' dissatisfaction.

Moreover, another disadvantage of stochastic programming is that the size of the resulting deterministic model increases drastically as a function of the number of scenarios, making the running time exponential. These two drawbacks are the reason why not everyone just uses the statistical approach, but why some researchers have tried to come up with alternatives such as robust optimization. Research has further been motivated by a recent increase in an demand for such an alternative due to volatile customer tastes, technological innovation and reduced product life cycles.

5 Robust optimization

The rest of this paper is going to be about robust optimization. Robust optimization is the approach that was chosen by Bertsimas, and his robust optimization method from 2002 is based on it. But Bertsimas was not the first person to develop such an approach. There were several predecessors.

5.1 Soyster's approach

Soyster's approach (1973) deals with uncertain, also called fuzzy, parameters in the matrix A. (1) The robust formulation of this approach is as follows:

$$
\begin{aligned}
&\max c'x \text{ where} \\
&\textstyle\sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij} y_j \le b_i \qquad \forall i \\
&-y_j \le x_j \le y_j \qquad \forall i \\
&l \le x \le u \\
&y \ge 0
\end{aligned}
\tag{2}
$$

J_i stands for the set of numbers indicating the coefficients that are subject to parameter uncertainty.

Instead of the simple set of inequalities $Ax \leq b$ this set of inequalities contains the more complicated set of inequalities $\sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij}y_j \leq b_i$, which means that another product is added. This product signifies the uncertainty. The values of y_j have to be chosen in such a way that for all i, the condition $-y_j \leq x_j \leq y_j$ holds.

We see that Soyster's approach considers column-wise uncertainty since there is one y_j for each column j.

The objective is to maximize the product $c'x$. For an optimal solution x^* the following property holds: $|x_j^*| = y_j$. For this reason, $\sum_j a_{ij}x_j^* + \sum_{j \in J_i} \hat{a}_{ij}|x_j^*| \leq b_i$ for all i must hold. Since $\sum_j a_{ij}x_j^* + \sum_{j \in J_i} \eta_{ij}\hat{a}_{ij}x_j^* \leq \sum_j a_{ij}x_j^* + \sum_{j \in J_i} \hat{a}_{ij}|x_j^*| \leq b_i$, the solution is provably robust: it works for all values a_{ij} may take.

Soyster's approach results in a linear optimization problem with $2n$ variables and $m + 2n$ constraints.

The drawback of Soyster's method is that in many cases, the obtained solution is not the best one possible. One may call Soyster's method very conservative: The detected solution is valid, no matter what value in the given range the coefficients in the matrix A take, but in many cases it is far from the optimum (which, however, would not be valid for any possible combination of parameters).

5.2 Ben-Tal's approach

The problem of over-conservatism which Soyster's approach exhibitis was addressed by Ben-Tal and Nemirovski in 1998. (3) (4) (5) Their approach allowed the uncertainty sets for the data to be ellipsoids. They proposed efficient algorithms to solve convex optimization problems under data uncertainty, which however lead to robust formulations that involve conic quadratic problems, and therefore these methods cannot be directly applied to discrete optimization.

In Ben-Tal and Nemirovski's approach, the formulation is modified as follows:

$\max c'x$ where
$\sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij}y_{ij} + \Omega_i sqrt(\sum_{j \in J_i} \hat{a}_{ij}^2 z_{ij}^2) \leq b_i \qquad \forall i$

$-y_{ij} \leq x_j \leq y_{ij} \qquad \forall i$
$l \leq x \leq u$
$y \geq 0$ (3)

We see from these formulae that Ben-Tal and Nemirovski's approach also has the advantage over Soyster's approach that now no longer only column-wise uncertainty is considered, but y becomes a matrix which specifies uncertainty for each element of the matrix a separately.

The probability that constraint i is violated is at most $exp(-\Omega_i^2/2)$ in this approach.

The resulting optimization problem is a conic optimization problem with $n+2k$ variables and $m+2k$ constraints.

5.3 Kouvelis and Yu's approach

Kouvelis and Yu proposed a framework for robust discrete optimization minimizing the worst case performance under a set of scenarios for the data. (2) However, many problems become NP-hard in that framework.

5.4 Bertsimas' approach

Bertsimas' approach from the year 2002 has a polynomial running time when applied to problems that are polynomially solvable in their original non-robust form. (6) Furthermore, it allows to control the degree of conservatism of a solution in terms of probabilistic bounds of constraint violation. This means that there is an additional parameter Γ_i which controls how many columns of the row i of the matrix a may take values in the given range; all other values are considered fixed.

This innovation has the effect that it is possible to find better solutions than using Soyster's approach. However, these solutions are less conservative, that is they are not feasible with all values the parameters in the matrix a can take.

The parameter Γ_i actually consists of two parts. The number before the comma signifies how many parameters may change in their full range, from minimum to maximum, while the number after the comma signifies how strong the deviation of one additional parameter may be. (For instance, $\Gamma_i = 8.36$ means that in constraint i, at most 8 parameters may differ from the lower up to the upper bound, and one ninth parameter may deviate by 36 percent of the total interval.)

The original, non-linear formulation of the optimization problem is as follows:

$$\max c'x \text{ where}$$
$$\sum_j a_{ij}x_j + \max_{\{S_i \cup \{t_i\}\}} \{\sum_{j \in S_i} \hat{a}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} y_t\} \leq b_i \qquad \forall i$$
$$-y_j \leq x_j \leq y_j \qquad \forall i$$
$$l \leq x \leq u$$
$$y \geq 0 \tag{4}$$

S_i is a subset of J_i that points to just as many coefficients as specified by the part of Γ_i before the comma. Another coefficient t_i may change by $\Gamma_i - \lfloor \Gamma_i \rfloor$ times the officially allowed deviation.

Bertsimas provides a conversion of this non-linear formulation to a linear optimization problem:

$$\max c'x \text{ where}$$
$$\sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j \in J_i} p_{ij} \leq b_i \qquad \forall i$$
$$z_i + p_{ij} \geq \hat{a}_{i_j} y_j \qquad \forall i, j \in J_i$$
$$-y_j \leq x_j \leq y_j \qquad \forall j$$
$$l_j \leq x_j \leq u_j \qquad \forall j$$
$$p_{ij} \geq 0 \qquad \forall i, j \in J_i$$
$$y_j \geq 0 \qquad \forall j$$
$$z_i \geq 0 \qquad \forall i \tag{5}$$

This model contains $n + k + 1$ variables and $m + k + n$ conditions, where $k = \sum_i |J_i|$.

It is possible to show that this approach not only finds an optimal solution if at most $\lfloor \Gamma_i \rfloor$ coefficients in row i of the matrix change, but also that there is a high probability that this solution is also valid in a general case.

The parameter Γ_i controls the trade-off between the probability that the solution is infeasibleand the effect on the objective function of the problem. Bertsimas calls this the "price of robustness".

A similar approach can be taken if the cost vector c is fuzzy. In this case, the objective $\max c'x$ is transformed to

$\max c'x + max \sum_{S_0|S_0 \in J_0, |S_0| \leq \Gamma_0} d_j |x_j|$

and this can also be converted to a linear formulation.

5.5 Example: Knapsack

As a demonstration how Bertsimas' approach to robust optimization works, the following chapter is a summary of the knapsack example from his paper from 2002. (6)

The original knapsack problem is a mixed integer program. One or several items whose weights are defined by the vector w have to be chosen. The total weight must not exceed b and the costs should be as large as possible.

$\max c'x$ where
$\sum_{i \in N} w_i x_i \leq b$
$x_i \in 0, 1$

A variant of this problem for which it would make sense to use robust optimization in order to solve it would be the one in which the weights are not exactly known, but only ranges are given. If S is a subset of the set of uncertain coefficients J containing $\lfloor \Gamma \rfloor$ values, t is a coefficient that does not appear in S, and δ_i are coefficients specifying the uncertainty of the weights, this new problem can be formulated as follows:

$\max c'x$ where
$\sum_{i \in N} w_i x_i + \max_{\{S \cup \{t\} | S \subseteq J, |S| = \lfloor \Gamma \rfloor, t \in J \setminus S\}} \sum_{j \in S} \delta_j x_j + (\Gamma - \lfloor \Gamma \rfloor) \delta_t x_t \leq b$
$x_i \in 0, 1$

Figures 1 and 2 show the effect of changing the size of the set J on the validity of the solution.

6 References

(1) Soyster, A. L. 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operational Research* 21 1154-1157.

(2) Kouvelis, P., Yu, G. 1997. Robust discrete optimization and its applications. *Kluwer Academic Publishers*, Norwell, MA

(3) Ben-Tal, A., Nemirovski, A. 1998. Robust convex optimization. *Math. Oper. Res.* 23, 769-805

(4) Ben-Tal, A., Nemirovski, A. 1999. Robust solutions to uncertain programs. *Oper. Res. Lett.* 25, 1-13

(5) Ben-Tal, A., Nemirovski, A. 2000. Robust solutions of linear programming problems contaminated withuncertain data. *Math. Prog.* 88, 411-424

(6) Bertsimas, D., Sim, M. 2002. The Price of Robustness. *Operational Research* Vol. 52, No. 1, pp. 35-53.

(7) Bertsimas, D., Sim, M. 2003. Robust discrete optimization and network flows. *Math. Program.* Ser. B 98: 49-71.

(8) William J. Cook 1997. Combinatorial Optimization. John Wiley and Sons.

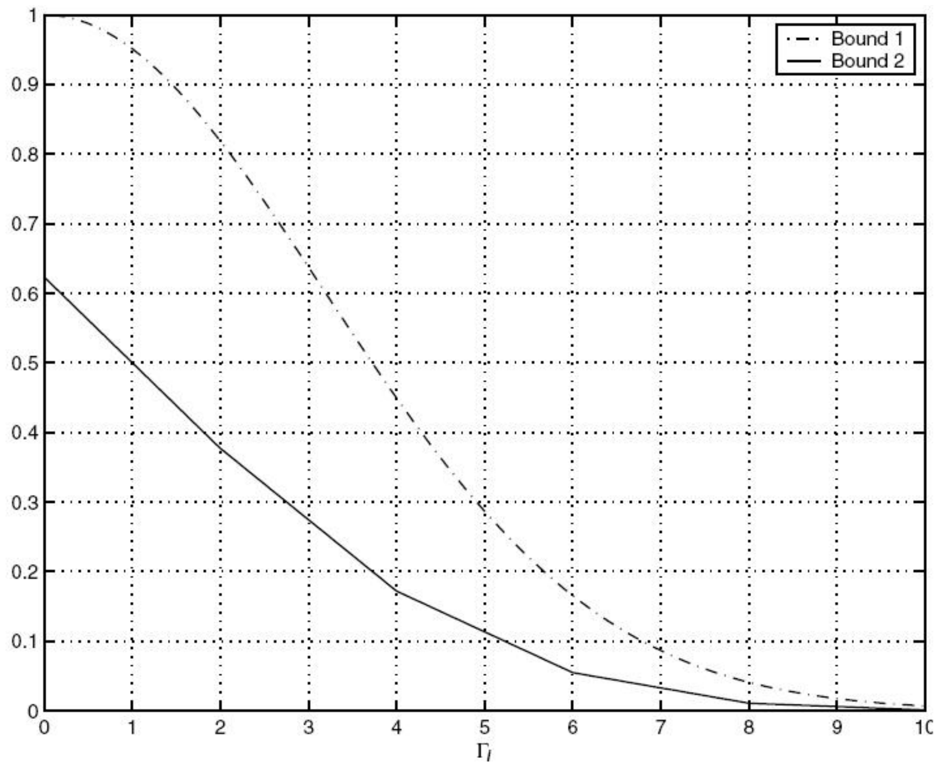

Figure 1: Probability of the validity of the solution for $n = |J_i| = 10$, from Bertsimas 2002

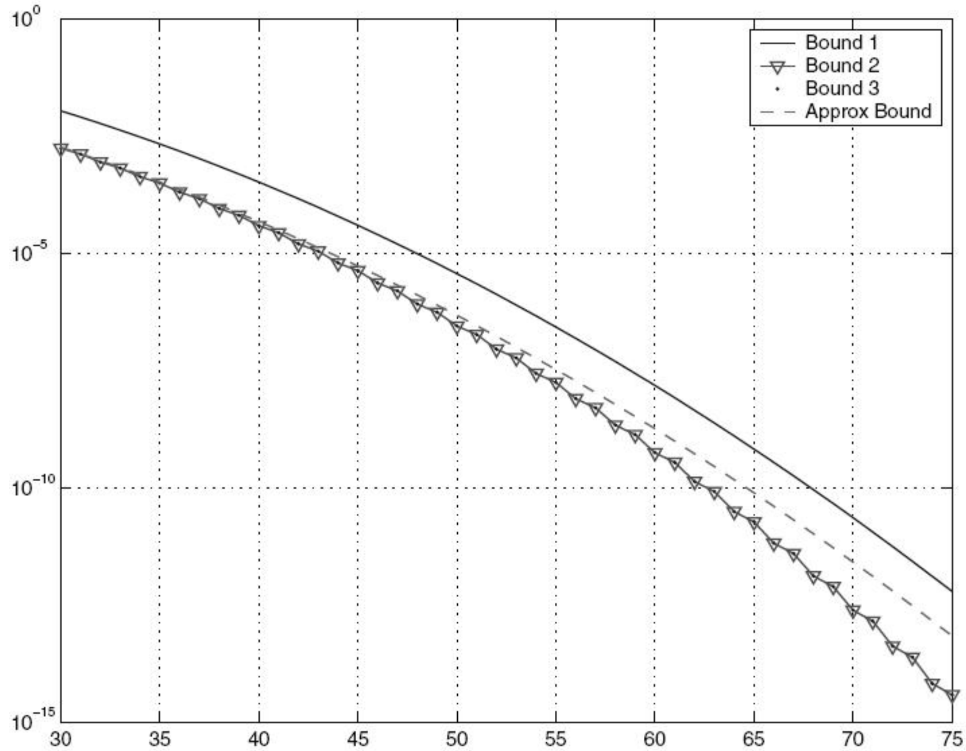

Figure 2: Probability of the validity of the solution for $n = |J_i| = 100$, from Bertsimas 2002