Een, twee, ..., ontelbaar

Inaugural lecture

Given on December 9, 2005
at Technische Universiteit Eindhoven

een, twee, ..., ontelbaar

prof.dr.ir. Karen Aardal

Introduction

Mijnheer de Rector Magnificus, Dames en Heren,
Combinatorische Algoritmiek. Dat klinkt goed! Een mengsel van informatica en wiskunde, wiskunde en informatica. Maar, wat is het eigenlijk? Wat wil ik doen met mijn vak? Wat zijn voor mij belangrijke vraagstellingen? Welke rol kan ik spelen aan de TU Eindhoven? Dit wil ik u in de komende 45 minuten proberen te vertellen. Omdat veel van de aanwezigen geen Nederlands spreken, zal ik mijn verhaal voortzetten in het Engels.

First I want to say that I am very excited to be affiliated with the Technische Universiteit Eindhoven. Ever since I, as a PhD student, made frequent visits to the Netherlands, Eindhoven was one of my favorite places to visit, not because of the beautiful city, but because of the University. For me TU/e, and then especially the Department of Mathematics and Computer Science, represents a combination of purely curiosity driven research and relevant and challenging applications.

My area
Before I proceed to some examples, let me briefly define my area in formal terms. The problems we study are *optimization problems*, and in particular *combinatorial optimization problems*, and for them we want to design and analyze *algorithms*. What is an optimization problem? In optimization we want to find a provably best solution among a given set of solutions. This set can be finite or infinite, and it is not given explicitly as a list of solutions, but *implicitly* as solutions to a set of equations and inequalities. Here we will consider *integer linear optimization* problems, which can be formulated as maximizing or minimizing a linear function subject to a set of linear constraints and the additional requirement that the variables should take integer values:

maximize cx
subject to $Ax \leq b$, x integer

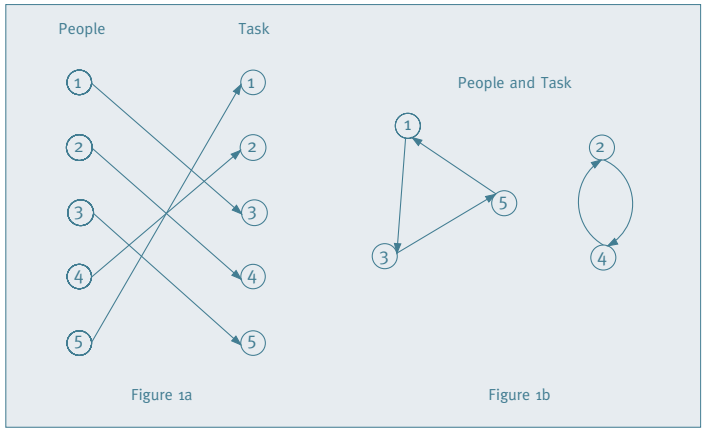
In *combinatorial optimization*, we look for a maximum- or minimum-

valued subset of a given set. This subset should satisfy certain properties. In the corresponding mathematical formulation given above, we typically have the restriction that the variables should take values zero or one.

What is surprising is that this simple formulation is such a powerful modeling tool. Determining frequencies for GSM networks, scheduling crews on flights, routing trucks to pick up and deliver goods at clients, scheduling trains at stations, designing communication networks, and determining the position of markers on a genome, problems in logistics, telecommunication and biology all fit in this modeling framework. We see people with a PhD degree in combinatorial algorithms and optimization go to different careers; from academic jobs, to large companies such as Philips, Dutch Railways, KLM, and KPN, and smaller consultancy companies such as Ortec and CQM. All these firms are obviously happy with the power and versatility of the modeling tool, but apparently also have effective means to solve these models. You might wonder: Is there anything left to do? The answer is a clear yes. I will try to highlight some relevant questions, but first, let us become more familiar with some problems.

figure 1

Two ways of representing a solution to an assignment problem.

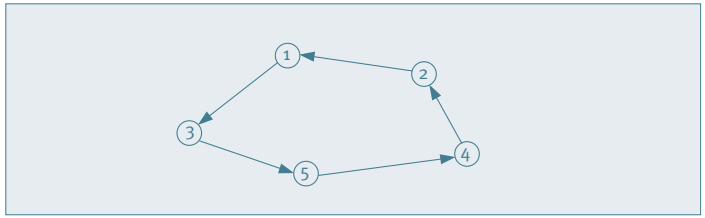


We begin with a problem that is simple to state. We have a set of tasks that need to be performed, and a set of equally many people who can perform them. Each person has assigned a cost to each of the tasks. The question now is how to assign precisely one person to each task such that the total cost of the assignment is minimized. This is the *assignment problem*. Is this an easy problem? Yes it is! Without going into any detail, there exists an efficient algorithm to solve it. A solution can for instance look as in figure 1a, but we can also draw the assignment as in figure 1b [1,2].

figure 2

As you can see in figure 1b, we have two cycles. Assume now that we restrict the solution space as follows. We do not wish to have several disjoint cycles, but we wish to find a single minimum cost cycle. This is the well-known *traveling salesman problem*.

A solution to the traveling salesman problem.



Is this problem easier or harder? One not so unnatural thought could be that, since we are actually restricting the solution space, it might be easier. After all, we exclude a lot of solutions, namely those consisting of disjoint cycles. But, the traveling salesman problem actually belongs to a class of problems, called NP-hard problems, for which we strongly believe that no efficient algorithm exists. So the traveling salesman problem is not easier than the assignment problem, and we *believe* that it is harder, but can we prove this? Not yet. The most prominent open question in my area, and in fact in all of algorithmics, is whether there is a distinction between easy and hard problems in the sense that some problems are efficiently solvable, and some *provably not*. This problem, called *P versus NP*, is listed as one of the seven *Millennium Problems* at the Clay Mathematics Institute Web page. It would earn you one million US Dollars if you could solve it. Who said that computer science and mathematics could not make you rich? In fact, I would be happy to pay one million dollars, and be poor, for being able to settle the question.

Solving combinatorial optimization problems

Just briefly reconnecting to the problem formulations: Both the assignment and the traveling salesman problems are combinatorial optimization problems. In both cases we want to find a subset of edges of a graph, and we can formulate them quite straightforwardly as zero-one optimization problems.

Are all NP-hard problems difficult to solve in practice? Most of the ones we know are difficult, but the variety in their actual difficulty is still huge, as I will try to illustrate next.

The traveling salesman problem and applications

Returning to the traveling salesman problem. How difficult is this NP-hard problem to solve in practice? It is certainly very difficult, but due, in particular, to advances in algorithms, but also to better software and faster computers, it is now possible to solve problem instances with about 25,000 cities to proven optimality (http://www.tsp.gatech.edu/) [3]. If we are satisfied with a very good, but not necessarily optimal, solution, then it is possible to obtain high-quality tours relatively fast for one million city-instances. It is worth noticing that a 25,000-city traveling salesman problem has about 312 million variables, so it is truly a huge problem.

How are optimal solutions computed? The naive approach would be to just enumerate all possible tours and pick the shortest one. After all, the number of tours is finite. How many tours do we have? Assume we have 10 cities in our problem. Standing at city one, we can choose between 9 cities to go to. After having made a choice, we can now choose between 8 cities, et cetera. So we obtain $9 \cdot 8 \cdot 7 \cdot \ldots \cdot 2 \cdot 1 = 362{,}880$ tours. Taking into account that it takes equally long time to traverse a given tour in the reverse direction, we actually get half as many tours, or 181,440. Well, we could certainly just enumerate all of them and choose the best. But, what about 20 cities? Such an instance has 1,216,451,004,088,320,000 tours, and a 40-city instance has 407,957,641,623,948,867,172,805,634,798,057,947,136,000,000,000 tours. So, you can easily imagine that it quickly becomes completely unrealistic to enumerate them all, once we get to interesting problem sizes. Before causing confusion, I want to point out that the sheer number of solutions is not really what makes the problem difficult – there are provably easy problems that have a larger growth of solutions than the traveling salesman problem – but it does exclude the possibility of doing complete enumeration.

I cannot explain in detail here how an algorithm works that solves the nontrivial traveling salesman instances, but I will explain the basic

principle. Assume we have found a feasible solution to the traveling salesman problem, that is, a tour. Is that difficult? Not really. A hungry monkey can do the job for you. Draw your problem on a large map. Put the map on the ground, and put a banana at each city on your map. The monkey will jump from city to city picking up bananas, and when he is done he has actually found you a tour, except that you might have to draw the final edge between the city where the exhausted monkey picked up the last banana, and the city where he started. But is that a good tour? Probably not. However, there are very good algorithms available for refining the 'monkey-tour', such that it really becomes a high-quality tour [4]. Since we are asking for the *shortest* possible tour, the length of any *feasible* tour is going to give you an upper bound on the optimum, that is,

length of optimal tour ≤ length of feasible tour.

So, a crucial question is: How can I actually determine whether or not a given feasible tour is an optimal one, without enumerating all feasible tours and comparing them? The way we do this is by providing a lower bound on the optimum that is equal to the upper bound. The main engine used in finding a lower bound is the so-called *linear relaxation*. We enlarge the solution space by simply ignoring the integrality constraints in our optimization problem. We allow fractional values of the variables, which means that the traveling salesman may split himself, going with, for example, one half of himself between city one and city two, and with the other half between city one and city three. For the traveling salesman problem, we have, through theoretical and algorithmic results, learned how to improve the initial linear relaxation, and how to extract relevant information from it. This enables us to produce an excellent lower bound that is used to prune the search process and to verify optimality relatively quickly in practice. Here, *relative* is impossible to make precise.

There are many reasons behind choosing the linear relaxation as the basis for a lower bound. One reason is historical; Dantzig [5] developed his simplex algorithm for linear optimization problems in the 1940's. When researchers then started to consider integer linear optimization problems, the linear relaxation was a natural starting point as it was a problem one knew how to deal with. Also, linear relaxation based

algorithms have been computationally very successful due to theoretical development, and this again has led to a continuing development of good software. This positive spiral is still ongoing. So, to summarize the status of the traveling salesman problem: We believe that it is theoretically hard, it is hard to solve in practice, but we are still pretty good at it, and we can really claim to solve practically interesting problem instances.

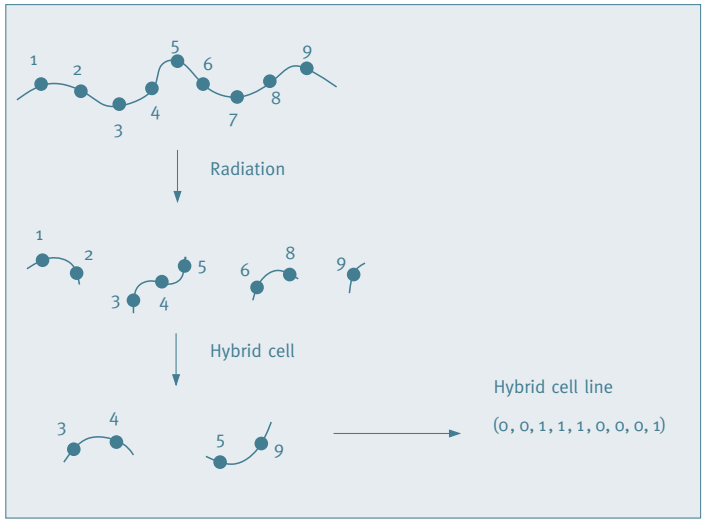
A relevant question at this point is: Who cares about traveling salesmen? Well, it is maybe unlikely that a traveling salesman computes his optimal route before stepping into his car, but this does not mean that there are no interesting applications. Given the season, I can mention one important application: How is Santa Claus finding the shortest tour visiting all children at Christmas Eve? This question was posed in the Florida Sun-Sentinel on December 20, 1998 on their Sunday Science page, see figure 4 and Bill Cook's TSP Web page: http://www.tsp.gatech.edu/.

Santa Claus finding the shortest tour visiting all kids.

figure 3

To be more serious, many relevant problems are essentially viewed as a traveling salesman problem. There are the relatively well-known ones, such as routing school busses, drilling holes in a printed circuit board, and routing trucks along warehouses and back to the depot. One not so obvious application comes from biology and the human genome project [6]. Here, given laboratory results from many sites, we wish to put these together so as to gather more, and better, information about the genome. In order to do this, we need to determine the accurate position of so-called *markers* on the genome. A marker is a DNA-segment that appears only once in the genome.

To determine the positions of the markers on a certain part of the genome, a lab uses a technique called *radiation hybrid mapping*. Here, the genome is subjected to high levels of x-rays, which makes it fall apart into fragments. Then, the fragments are combined with genetic material from rodents to form hybrid cell lines. These hybrid cells are then analyzed for the presence of markers. This is illustrated in figure 4. This experiment is repeated a number of times so as to obtain several different cell lines.



The principle of radiation hybrid mapping.

figure 4

If two markers are close to each other on the genome, they are more likely to appear in the same hybrid cell line than if they are far apart on the genome. This is used to define a distance function on the markers.

Each cell line that is obtained after this experiment can be represented as a string of zeros and ones. As you can see in figure 4 we have obtained the cell line (0,0,1,1,1,0,0,0,1) where the ones are at the positions of the markers that belong to that cell line, in this case in positions three, four, five, and nine. Assume we have conducted five of these experiments and obtained cell lines as follows:

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

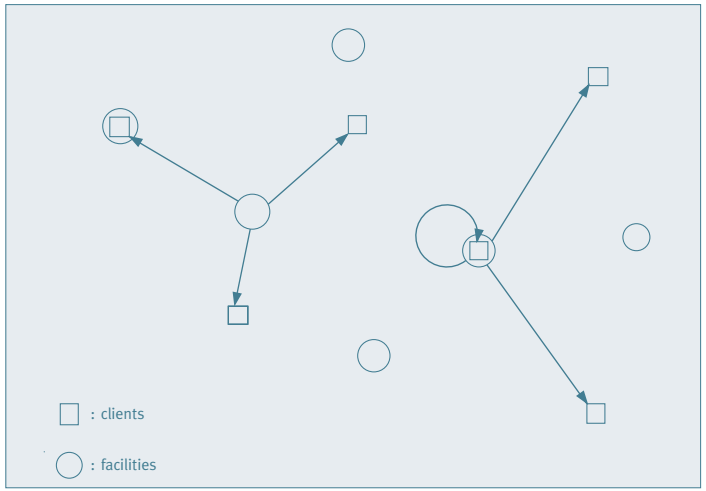
Each column of this matrix represents a marker, and each row a cell line. The distance between two markers is defined as the number of positions that their columns differ. Look for instance at the two last columns. They differ in two positions, making the distance between them equal to two. To find the ordering of the markers in the investigated part of the genome, we just find a shortest tour through the markers.

The computer code CONCORDE [3], which has been used to solve the largest traveling salesman problem instances to optimality, is also used by the National Institute of Health in the USA to build radiation hybrid maps, and this approach has led to improvements in the quality of the maps and in the speed with which they are obtained. The salesman has also been traveling through markers of other species, among them: horses, dogs, cats, mice and rats.

An easy 'hard' problem

Let me give you another example of a combinatorial optimization problem, called the *facility location problem*. Here we have a set of clients with a demand of a certain type of goods or services. The goods or services are provided by facilities, and we have a given set of potential location sites for these. There is a fixed cost associated with establishing a facility at a given site, and for each pair of client and potential facility there is an assignment cost. So, the question is: Which facilities should we open, and to which open facility should each client be assigned such that the total cost is minimized? An example of an instance and a solution to this instance is given in figure 5.

This is also an NP-hard problem. Is this easy or hard in practice? That depends on the objective function. If we assume that the fixed costs and the assignment costs are really costs, that is, they are nonnegative, and if the assignment costs are basically the distances between the corresponding facilities and clients, then the problem is very easy to solve. What often happens is that the linear relaxation produces an integer solution immediately, and if not, the search is typically very brief. So, in practice the problem is easy. How come? This is something we do not completely understand, but it is clear that for some reason, the linear relaxation very closely approximates the set of integer solutions in the part of the solution space that is relevant for this type of objective



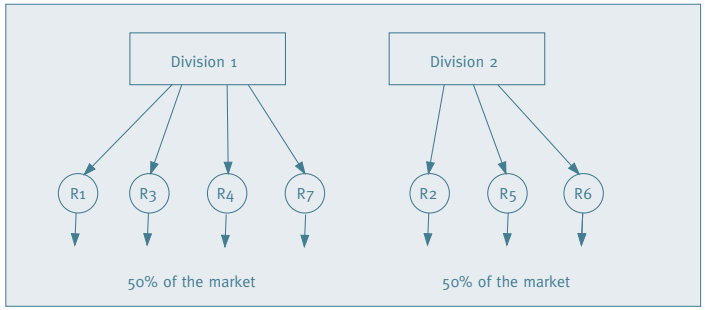
An example of a solution to a facility location problem.

figure 5

function. If we change the objective function to also allow for arbitrary costs, then the problem suddenly becomes considerably more difficult to handle.

A hard 'hard' problem

I will introduce one final combinatorial optimization problem to you. A version of it was described already in 1978, in the book on mathematical modeling by Williams [7], and later in 1998 in a specific version by Cornuéjols and Dawande [8]. The problem is called the *market share problem*, and is described as follows. A retail company has created two divisions and wants to assign each retailer to one of the divisions. The assigned division will then supply the retailer with products that the retailer sells to the customer. For each retailer we know how much of the different products he estimates to sell, and we also know the total demand of each product. The question now is: Is it possible to assign the retailers to the two divisions such that the divisions obtain a given prescribed share of the market?



The market share problem.

figure 6

This is a *feasibility problem* rather than an optimization problem, and it is NP-hard. The Cornuéjols-Dawande instances of the market share problem were generated for the purpose of demonstrating that there are still small combinatorial optimization problems that are very hard to solve, and to provide benchmark instances for testing new algorithms for this class of problems.

In contrast to the traveling salesman problem, which is difficult but where we can still tackle instances with ten thousands of cities, the market share instances become difficult to solve already for 40 to 50 retailers. And, some of my colleagues speculate whether we will ever be able to solve instances with 100 retailers. Why is the market share problem so difficult? This again is not very well understood, but if we look at the mathematical formulation of the Cornuéjols-Dawande instances, we at least get a first hint. We have one variable for each retailer, and one constraint for every product. The variables should all take value zero or one and the constraints are all linear equations. We have about ten times as many variables as the number of equations, and we ask for a 50-50 split of the market between the divisions.

A linear relaxation is feasible, and likely to remain feasible if we fix many assignments of retailers to divisions. Hence, even deep in the search tree we get poor information from the linear relaxation, and we are practically bound to perform complete enumeration. Vaguely said, infeasible instances are not infeasible enough to be recognized quickly, and feasible ones contain so few integer solutions that we will not find

any easily. And, we do not know how to enrich the information produced by the linear relaxation.

Again, the naive approach will not work either, even though we have few variables. Complete enumeration means enumerating all 2^n potential assignments of retailers to the divisions, where n is the number of retailers in our model. For $n = 50$ we have 1,125,899,906,842,624 potential assignments, and for $n = 100$, there are

1,267,650,600,228,229,401,496,703,205,376

potential assignments. How do you even call this number? It depends on whether you use the US or British nomenclature. In the US system it is called one nonillion, 267 octillion, 650 septillion, 600 sextillion, 228 quintillion, 229 quadrillion, 401 trillion, 496 billion, 703 million, two hundred and five thousand three hundred and seventy six! How about that! This is certainly more than we can handle using any futuristic computer system.

This leads me to a small sidetrack. When my daughter Liesje is wondering about a large number, she asks me (in Dutch): “Is het 10?” I say no, “Is het 100?” Nope. “Is het een miljoen miljard triljard ontelbaar?” So, the concept of large numbers, and even infinity, is apparently present also at a younger age. But there are people who only basically count to two. Anything above that becomes ‘many’. Paul Vitányi [9], a colleague at CWI, is writing a book about counting. He kindly made a chapter available to me. I am citing:

“There are natives in Queensland who count ‘two, two-one, two-two, much’, and the Bushmen of Botswana count ‘a, oa, ua, oa-oa, oa-oa-a, and oa-oa-oa’ up to six, and then use a word meaning ‘many’. More restricted are the Damara of Namibia who count ‘one, two, many’.”

In this context I guess we can safely say that we have *many* potential solutions to our problem.

Is there an alternative to using the traditional linear optimization based enumeration approach? Well, one is tempted to say: There has to be. I will illustrate one idea that at least works better than the traditional approach. This is not to say that it is the ultimate way of solving the market share instances, but it illustrates that by viewing the problem in

a different way, we can get further. In this case to at least to 70 variables, which is at least a step forward. To illustrate the idea I will use an even more 'practically hard' problem.

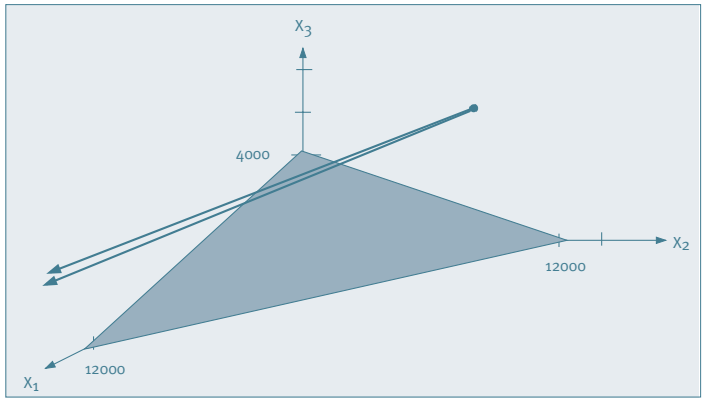
The integer knapsack problem

This is structurally speaking one of the cleanest problems around, but still NP-hard. We want to pack items in a 'knapsack'. We know how much weight each item has, and we can pack an integer nonnegative number of each item. The knapsack has to be packed to an exact given positive integer weight. So, we have to satisfy one equation in integer nonnegative numbers. It is also sometimes referred to as the 'coin exchange problem'. Can we combine coins of different monetary value to reach an exact given value? It is interesting to notice that if we forget about nonnegativity, then the problem can be efficiently solved. Consider the following specific example:

$$12{,}223\, x_1 + 12{,}224\, x_2 + 36{,}671\, x_3 = 149{,}389{,}505$$

If we for instance pack 12,222 units of x_1, we get one unit too many. Or, if we take 12,221 units of x_2 we get one unit too few. But maybe there is a more sophisticated combination of the different items such that we get bang on the target? If we view this geometrically we see that the linear relaxation is a large triangle. So the question is: Is there an integer point in this large triangle? Since we know how to efficiently determine whether there is an integer point in the *same plane* as the triangle, this seems like a good starting point. Here we assume that such an integer point exists, and once we have found it, we use it as the origin of a new two-dimensional coordinate system, see figure 7. We of course have to be careful and make sure to create a coordinate system that gives us a correct description of all the integer points in the plane. This can again be done efficiently.

The first question that comes to mind is: How many different coordinate systems exist that correctly describe all the integer points in this plane? The answer is: Infinitely many. So, there is an issue of finding a 'good' one, and then we need to think about how we measure 'good', and whether we can determine a 'good' one efficiently. Well, we begin with the system we see in figure 7, and then we check later to see if it looks good. To make the picture more clear, I have also drawn the triangle in the new two-dimensional coordinate system in figure 8.



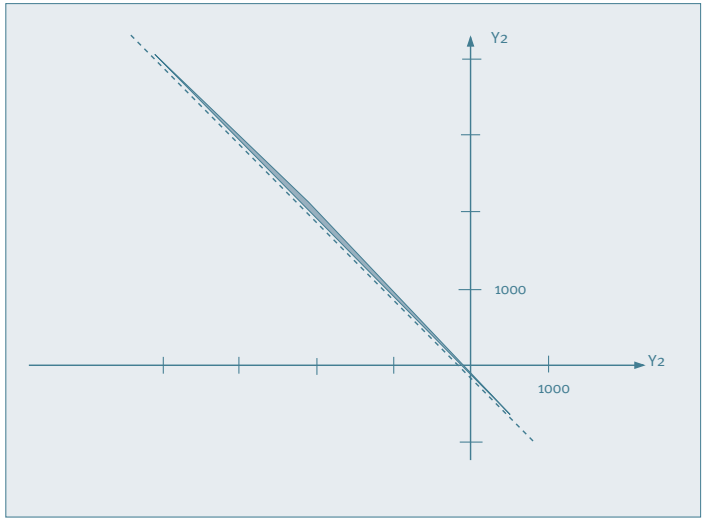
A new two-dimensional coordinate system.

figure 7

We observe that our triangle is directed diagonally in the new system, which is likely to confuse our traditional search methods. The triangle goes far in both coordinate directions, so there are many potential integer points in it or close to it. If we had drawn the picture in much more detail, we could have *seen* that the triangle nicely 'cruises' *between* the integer points, but does not contain any. Can we verify this *algorithmically*? This is equivalent to asking whether we algorithmically can identify an integer orientation, or direction, in which the mapped triangle is thin. And, the answer is, yes, it is possible [10,11]. In fact, the algorithm I am referring to produces an integer direction efficiently that has the following property. Suppose we take parallel lines in the integer direction that the algorithm produces, such as the dashed line in figure 8, and suppose we put them in our coordinate system such that they cover all integer points. Then, at most c of the lines intersect our triangle. Here, c is a constant that depends only on the dimension of the space we are working in. The algorithm is beautiful, but still quite demanding computationally.

Let us return to the question whether our coordinate system looks good or not. As you can see back in figure 7, the angle between the coordinate axes is acute. The consequence is that the triangle mapped in this system has a diagonal orientation as in figure 8. Can we do better? Yes! Can we do it better algorithmically? Yes again. This can be done, efficiently, by

the Lenstra-Lenstra-Lovász basis reduction algorithm [10]. This algorithm guarantees an 'almost' orthogonal coordinate system with relatively short vectors. So, let us use this algorithm and look at the outcome.



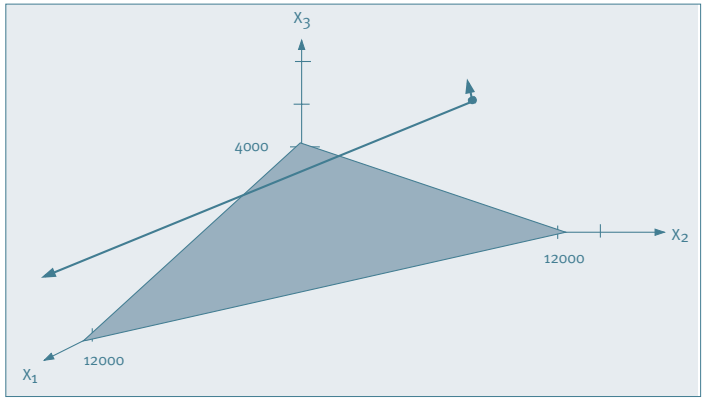
Our triangle in the new two-dimensional coordinate system.

figure 8

In our case the algorithm in fact produced an orthogonal system. The new short vector in figure 9 is the difference of the vectors in figure 7. The second vector is still long, but it cannot be shortened further: its length simply reflects the structure of the input. The orthogonality of our new coordinate system means that the mapped triangle will also be oriented orthogonally in the two-dimensional system. As we see in figure 10, the triangle is now nicely located strictly between two lines that are unit directions in our new system. This immediately provides us with a certificate that the triangle does not contain any integer point. Notice the different scale on the two coordinate axes. It is of course not true in general that we get such an extreme outcome, but it is true that certain instances become substantially easier to solve if we take this approach [12,13]. Among them are the market share instances and integer knapsack instances where large multipliers are 'hidden' in the input.

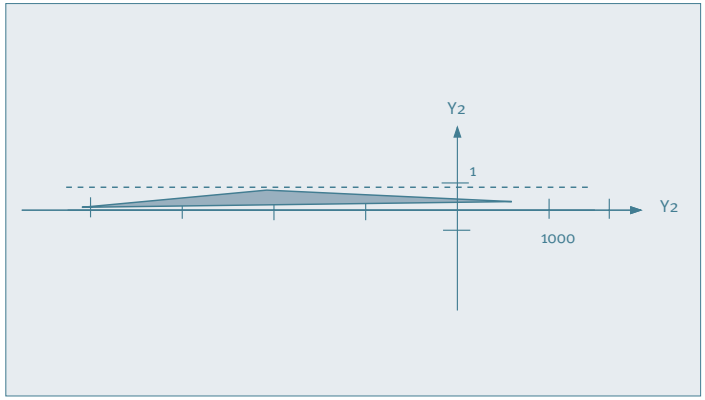
figure 9

If we solve our 3-variable example using a traditional search algorithm that relies on the linear relaxation, we need more than one million iterations. If we extend this example into more than, say, dimension 5, we can simply not solve them at all at this point in time using standard methods, whereas our coordinate transformation makes them trivial.



A second two-dimensional coordinate system.

figure 10



Our triangle in the second coordinate system.

Some final questions

As you have seen, NP-hardness seems quite a crude measurement on what to expect regarding practical solvability; some problems are hardly solvable in dimension 5, whereas others seem to have no practical limit. This is a topic that my algorithms colleague Mark de Berg also treated in his inaugural lecture. I find it a very challenging question to try to better understand why some problems seem so practically intractable and some so much easier than others. Is there a way of predicting this? Or, maybe the problems that seem so hard are actually quite easy once we understand them better? How much is a structural, and how much is an algorithmic problem? Where is the lack of understanding?

Educational and societal aspects

Even though I have only had time to demonstrate a few problem types, I still hope you can appreciate the potential applicability of optimization and the existence of interesting structural and algorithmic questions. The area intersects many other areas, such as algebra, number theory, combinatorics, and graph theory, to mention a few, which also means that it heavily makes use of theory from these areas. So, to keep continue to bring in new ideas to the field we depend on people with a broad but also quite a deep knowledge. This leads me in the direction of teaching and research policies. In the following I speak on a general level and do not in particular refer to the situation in Eindhoven. Nor do I refer only to my own area, but more in general to areas that depend on a certain level of mathematical knowledge.

One thing I find sad is that most people do not realize that computer science and mathematics are around us every day in a very positive sense. They think of a computer as a black, or sometimes boring beige, box, and of a weather forecast as something you listen to after the news. But few people actually reflect upon that this would not at all have been possible without computer science and mathematics. The general public thinks, especially of mathematics, as something for a very select and slightly weird group. If you do not believe me, try to tell your hairdresser, or the person next to you in an airplane, what your profession is and see how enthusiastically the conversation continues. When I am asked, and I tell what I do, this is a definite conversation stopper. Given this image, it is not so easy to attract potential students even though the market situation for people with a computer science or mathematics education is brilliant. How can universities influence this image? This leads me to an article in the Dutch *Nieuw Archief voor Wiskunde* from 2003, by the past president of Utrecht University, Jan Veldhuis. He was invited to give a presentation at the symposium “Wiskunde, nodig en in nood” at the Dutch Mathematics Congress. He said, and this is my translation:

“Which possibilities do we have to make sure that mathematics stays present at our universities? ... The current percentage of students who

complete a mathematics program, 45%, has to be increased. We have to keep offering enough to the especially talented and ambitious students who are interested in mathematics in itself, but not all students have to study mathematics in depth; 25% is sufficient and acceptable. The rest can be offered a broader and a more differentiated program through connections with computer science, biology and economics."

I very much agree with this. I would like to see more application areas treated *within* the mathematics program, and also within the computer science program. Not at the expense of the 'pure' topics, but in addition. In this way we can make the wide applicability of computer science and mathematics more transparent, and we also ensure that the students obtain a rigorous foundation that will keep new ideas and approaches enriching the application areas. In this context it is also interesting to observe that some of the most high-profile applications in my area have been carried out by people who first became well known for their theoretical accomplishments. I am optimistic that a positive change will occur with the bachelor-master system. And, all new activities that are initiated with the creation of for instance various mathematics clusters should have a positive effect. In fact you are attending a DIAMANT-cluster event right now!

A final issue

I hope you will bear with me when I spend two minutes on the gender issue, being a gender issue myself. A woman who is pursuing an academic career in the Netherlands has a hard time. Role patterns of the past are still obstacles today. At one of my previous academic employers in this country, the secretaries would happily type letters for male PhD-students, not for female professors. As a results of my open-door policy, delivery men would pile up new loads of xerox paper in my office, no questions asked, and no apologies offered.

Not so long ago, dear mr. Rector, your photographer was going to shoot the picture for the invitation for today. Armed with his equipment he walked past my office. I got into the corridor, "I believe we have an appointment?" "No, I'm not looking for a girl, I'm looking for Professor Aardal." Oh, well. Once he had completed his setup, he asked, "Can you give me an intellectual look please?" Which was a far call at that point. So much for equality of the sexes in Dutch academia.

These are of course harmless examples. More difficult to tackle are the more subtle hurdles that women do encounter, especially if they have young children. You need extra confidence to keep going against the cultural pressure. I hope my presence today might stimulate some young female faculty to muster the courage for the extra step that is needed. Think about what Johan Cruijff said: "Je moet schieten, anders kun je niet scoren". Next to all the beauties that algorithms for integer optimization has to offer, this item will be on my agenda. And on yours, I trust.

Acknowledgements

I wish to end this lecture with thanking several people who have had a very positive influence on me, personally and professionally. First I wish to thank the Technische Universiteit Eindhoven, and the Department of Mathematics and Computer Science for their trust in me. Mark de Berg, Gerard Woeginger, and Leen Stougie played an active role in making my appointment possible, and the Department administration represented by Hans van Duijn, now our rector, Kees van Hee, and Suzanne Udo, made me feel very welcome. My appointment date, April 1, initially made me feel worried about the seriousness of the appointment, and receiving my contract from a rector with the same appointment date was not reassuring. But, we are as far as I know both still under contract, which is a good sign.

For me, returning to the Netherlands after almost two academic years at Georgia Tech was not entirely easy. My colleagues at CWI, Bert Gerards, Monique Laurent, and Lex Schrijver, have been generous to me, and I am grateful for their support and friendship, and for the stimulating and extremely high-quality environment they are creating. The group of PhD students and postdocs keeps us young and alert, and makes the professional environment complete.

Everyone can probably point out a pivotal moment in his or her career. For me it was the possibility of studying for my PhD at CORE with Laurence Wolsey as my supervisor. Laurence taught me, and is still teaching me by example, the importance of asking the right questions. His scientific integrity also serves as an important example to me. Dear Laurence, thank you for all the time and effort you have invested in me and for your continuing friendship. Next to Laurence, Yves Pochet always supported me immensely during my time in Belgium. I know few with such a good eye for problem structure and solution.

Over the years I have had the privilege to work with and learn from many very impressive and nice colleagues. It is impossible to thank them all individually, so I hereby direct a big thank you to each and every one of you!

The most important support comes from your family. My parents stimulated me to make my own academic choices, and my father's work on several aircraft engines stimulated my interest in science. I wish he could have been here today. My mother is here, together with some of my extended family members. Thank you for your support and for making the long trip to be with me here today.

Finally, dear Jan Karel, you have provided me with support and good advise since we first met. You are always positive, honest, and encouraging, and your attitude forms an ideal counterpart to my more hesitating self. Moving around a lot has been great fun, but with you, Liesje en Jacob, and Catrien and Hidde, I feel at home.

Dames en heren, bedankt voor uw aanwezigheid en aandacht. Lieve Liesje en Jacob, bedankt voor jullie geduld!

Ik heb gezegd.

References

1 R.E. Gomory, The Traveling Salesman Problem. In: *Proceedings IBM Scientific Computing Symposium Combinatorial Problems,* IBM, White Plains, NY, 1966, pp. 93 - 121.

2 L.A. Wolsey, *Integer Programming,* John Wiley & Sons, Inc., New York, NY, 1998.

3 Bill Cook's home page contains a substantial amount of information on the traveling salesman problem, including historical information, http://www.tsp.gatech.edu/.

4 David Johnson's home page is an excellent source for information on finding feasible traveling salesman tours. See especially the part on the 8th DIMACS Implementation Challenge, http://www.research.att.com/~dsj/chtsp/index.html.

5 G.B. Dantzig, Maximization of a linear function of variables subject to linear constraints, In: Tj.C. Koopmans (ed.) *Activity Analysis of Production and Allocation*, Wiley, New York, NY, 1951, pp. 339 - 347.

6 Bill Cook, private communication, and the paper: R. Agarwala, D.L. Applegate, D. Maglott, G.D. Schuler, and A.A. Schäffer, A fast and scalable radiation hybrid map construction and integration strategy, National Center for Biotechnology Information, Bethesda, MD.

7 H.P. Williams, *Model Building in Mathematical Programming,* Wiley, Chichester, 1978.

8 G. Cornuéjols and M. Dawande, A class of hard small 0-1 programs, *INFORMS Journal on Computing* **11**(2), 205 - 210, 1999.

9 P. Vitányi, *Counting: Number Representations and Counter Machine Algorithms*, Wiley-Interscience Series in Discrete Mathematics. In preparation.

10 A.K. Lenstra, H.W. Lenstra, Jr., and L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen* **261**, 515 - 534, 1982.

11 H.W. Lenstra, Jr., Integer programming in fixed dimension, Mathematics of *Operations Research* **8**, 538 - 548, 1983.

12 K. Aardal, A.K. Lenstra, Hard equality-constrained integer knapsacks, *Mathematics of Operations Research* **29** (3), 724 - 738, 2004.

13 K. Aardal, R.E. Bixby, C.A.J. Hurkens, A.K. Lenstra, and J.W. Smeltink, Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances, *INFORMS Journal on Computing* **12**, 192- 202, 2000.

Curriculum Vitae

Prof.dr.ir. Karen Aardal is per 1 april 2005 benoemd tot deeltijd hoogleraar combinatorische algoritmiek aan de faculteit Wiskunde & Informatica van de Technische Universiteit Eindhoven (TU/e).

Karen Aardal studeerde aan Linköpings Tekniska Högskola, Zweden, met als afstudeerrichting Besliskunde. Na haar afstuderen in 1985 legde ze in 1988 bij dezelfde universiteit een Teknisk Licentiat examen af. In 1989 begon ze haar promotieonderzoek op het gebied van de polyhedrale combinatoriek aan de Université Catholique de Louvain, dat ze in 1992 afrondde. Zij is werkzaam geweest bij de University of Essex (Engeland), de Erasmus Universiteit, de Katholieke Universiteit Brabant, de Universiteit Utrecht en het Georgia Institute of Technology (VS). In maart 2004 trad ze in dienst van het Centrum voor Wiskunde en Informatica in Amsterdam als senior onderzoeker in de groep Networks and Logic – Optimization and Programming.
Karen Aardal richt zich in haar onderzoek voornamelijk op algoritmen voor geheeltallige optimaliseringsproblemen, waarbij ze gebruik maakt van structuren van bepaalde roosters en van verscheidene relaxaties. Daarnaast heeft zij veel belangstelling voor toepassingen in de logistiek en de telecommunicatie.

Colophon

Production:
Communicatie Service
Centrum TU/e

Photography cover:
Rob Stork, Eindhoven

Design:
Plaza ontwerpers, Eindhoven

Print:
Drukkerij Lecturis, Eindhoven

ISBN: 90-386-1523-X

Digital version:
www.tue.nl/bib/