Yugoslav Journal of Operations Research
5 (1995), Number 1, 3-19

THE USEFULNESS AND BEAUTY OF COMBINATORIAL OPTIMIZATION

Jens CLAUSEN, Jakob KRARUP

DIKU, Department of Computer Science, University of Copenhagen, Denmark

Abstract: Disregarding combinatorial optimization as an attractive platform for academic careers, the three main justifications for the current and steadily increasing interest in the field are:

- the variety of realistic decision problems amenable for modelling and analysis via combinatorial optimization
- the lack of a universal, operational algorithm
- other theoretical challenges

First the nature of a combinatorial optimization problem is accounted for. To substantiate the significance of such problems to decision-makers in practice, an overview of the most profitable application areas as well as the most applicable problem types is then provided. Past history is briefly reviewed within the framework of an annotated bibliography. We close with subjective views as to today's challenges and to what is believed to be tomorrow's main issues of concern.

Keywords: Combinatorial optimization, enumeration, heuristics, theoretical challenges.

1. COMBINATORICS AND COMBINATORIAL OPTIMIZATION

One of the main concerns of mathematicians engaged in *combinatorics* is that of counting. For a given set of objects, how many of these objects possess certain properties? For example, among all graphs with p points, how many of these are nonisomorphic trees? Such *enumeration* problems, however, do not only arise in pure mathematics but are also encountered in other branches of sciences as well as in everyday life. A bridge player contemplating possible strategies must estimate certain probabilities, that is, certain ratios between all outcomes in favour of some event and all possible outcomes. Unless the estimate is based solely on experience and intuition a thorough analysis must inevitably imply enumeration.

Another key issue in combinatorics that of *existence*. Does a given graph contain a cycle passing through each vertex exactly once? For general graphs, what are necessary and sufficient conditions for the existence of such a cycle? Another famous example is the following question: does there exist two trees with exactly the same number of leaves? The affirmative answer, at times attributed to the Danish 19th century philosopher, Søren Kirkegaard, is: "Since the total number of trees exceeds the largest number of leaves on any tree, then there must be at least one pair of trees having exactly the same number of leaves". Note, however, that this indisputable proof of existence gives no clues whatsoever as to *how* such a pair of trees actually should be found.

The aim of combinatorial optimization is different. *A combinatorial optimization problem* (COP) can be stated as follows. Given a finite or countably infinite, discrete set S of possible actions. With each action $x \in S$ is associated a real-valued function $f(x)$. Find a possible action minimizing (or maximizing) $f(x)$, that is, $\min \{ f(x) : x \in S \}$. As an example, consider the following routing problem, known as the *symmetric traveling salesman problem* (STSP): Let n cities and the distance between each pair of these be given. Find a tour of minimum length, starting and ending at some city, and visiting all other cities exactly once. Here, x is a specific such tour of length $f(x)$, and S is the set of all $\frac{1}{2}(n-1)!$ distinct tours. STSP itself is an example of a *problem* or a *problem type*, that is, an example of a general question to be answered, usually in terms of assigning specific values to a set of *variables* such that the resulting solution satisfies certain properties. Of course, algorithms for solving a well-defined problem can be proposed regardless of specific values of the problem's *parameters*; for STSP these consist of the number n of cities, and each of the $\binom{n}{2}$ intercity distances. Whenever all such parameter values are specified, we talk of a *data instance* or just an *instance* of the corresponding problem. In turn, a problem is occasionally defined as the class of all instances of a specified form.

The two constituents of a COP, the criterion function $f(x)$ to be extremized (max or min) and the set S of possible actions or feasible solutions call for several comments. $f(x)$, which is supposed adequately to reflect the decision-maker's preferences, will typically represent profit (to be maximized) or total costs (to be minimized). Depending on the nature of the set S, two distinct groups of COPs can be distinguished. Most often S is determined by a set of linear equations and equalities to be satisfied simultaneously with the additional requirement that some or all variables must be integer. Occasionally, S is a subset of a large finite set of potential feasible solutions of which the feasible ones constituting S satisfy additional constraints. As will be elaborated upon later on, the computational techniques available for solving a given COP are heavily dependent upon the structure of S. Both representations of S, however, can appropriately reflect the inherent indivisibilities characterizing phenomena in the real world (canned sardines to be shipped, aircrafts to be scheduled, bridges to be built, factories to be located, personnel to be assigned to jobs, etc.) and are in themselves plausible explanations for the strong interest in combinatorial optimization. Rather obvious cases of the "shipping canned sardines" – type do occur in practice, but they are not common in comparison with decision problems boiled down to a series of go/no go, locate/not locate, either/or questions. Actually, the real power of the field as a modelling tool is its ability to handle decision problems where all variables are restricted to the two values, 0 or 1.

0–1 variables do not exclusively represent either–or decisions; they are also widely used as means of modelling, for example, logical conditions, and non–convex feasible regions. Almost all of the models listed in the sequel and ranked as the "most applicable" models of combinatorial optimization involve 0–1 variables only.

As to the solution of COPs from the first group, one straightforward approach is to *relax* the integer requirement (that is, for example, to replace "$x_{17} \in \{ 0, 1, 2, 3, 4 \}$" by "$0 \leq x_{17} \leq 4$" and likewise for all other integer–valued variables), and then to solve the resulting *LP–relaxation* of the original COP. Rounding off the optimal LP–solution values to nearest integers may lead to an acceptable solution to the underlying COP if these integers are relatively large. On the other hand, it is easy to construct examples involving only two 0–1 variables where the result of such an approach is "arbitrarily bad".

Assuming that the cardinality $|S|$ of S is finite (which by definition is true for COPs in the second group), another solution procedure guaranteeing optimality in a finite number of steps is that of *complete enumeration*: evaluate $f(x)$ for each of the $|S|$ feasible solutions and pick the best. As the time needed for this task roughly is proportional to $|S|$, we are in such cases indeed interested in knowing the total number of feasible solutions.

Even for modest–sized data instances of COPs encountered in practice, however, $|S|$ is normally an astronomical number, orders of magnitudes larger than the total number of elementary particles in the universe. Yet, for example, most papers dealing with STSP commence with the canonical sentence, "For STSP with n cities, the number of distinct tours is $\frac{1}{2}(n-1)!$". Moreover, some authors like to state, say, the 19–digit long exact value of $20!/2$ and to mention that a solution approach based on complete explicit evaluation of the length of each tour would require an exorbitant amount of CPU–time, here, on the order of thousands of years, even on the fastest computer available. A more sophisticated, but, from a computational viewpoint, equally useless statement is that the number of feasible allocations of clients to facilities in certain locational decision problems is expressible in terms of the so–called Stirling numbers of the second kind.

The conclusion drawn at this stage are: (1) Issues of enumeration and existence provide insight but are of limited value to the development of algorithms. (2) Cases do occur (for example, when the problem essentially is to analyse consequences based on a handful of scenarios) where $|S|$ is so modest that the use of complete enumeration can be advocated. (3) Even exorbitant numbers of feasible solutions, however, should not leave us with an impression of COPs as being computationally intractable. On the contrary, in assessing the computational complexity of some COP, the cardinality $|S|$

may in general be quite misleading. There is a wealth of COPs and corresponding large-scale data instances for which cleverly designed algorithms can identify an optimal solution in matters of seconds. On the other hand, there are also optimization problems for which the largest instance that can be solved to optimality by the "best" algorithm available is surprisingly small.

The distinction between a combinatorialist engaged with counting and existence and an optimizer searching for a "best" solution need not be sharp. In order to estimate the expected time to be spent with the job, a burglar in front of a safe must have some idea as to the number of combinations to be examined. His preference function $f(x)$, on the other hand, is rather special in that only a single action is preferred to the remaining $|S|-1$ actions among which he is totally indifferent. Those who dislike the flavour of this example may instead think of unlocking a bike (normally also under time pressure) when the right combination has escaped the mind.

The burglars-in-front-of-a-safe example can be pushed a step further and yet remain highly realistic. We are here referring to cases where the set S of feasible solutions in a sense is unknown (and possibly even empty) and where the value $f(x)$ is the same for all $x \in S$. Such a situation has, for example, been encountered in a recent case study dealing with the schedule of various activities at the Royal Theatre, Copenhagen. All feasible solutions were ranked equally as the overall objective simply was to identify a single solution satisfying a very complex set of constraints.

In some respects, life was relatively easy for consultants and decision-makers in the "happy sixties" where the choice among alternative actions was based solely on a single criterion like "maximize profit" or "minimize total costs". Such simplistic measures of performance are employed in almost all of the "most applicable models" listed in the following section and reflect the concurrent development of OR and computer science of that time. The various crises experienced later on and the changing values, however, have led to an increasing interest in the ability to operationalize the handling of more complex criteria. Such models for *multiple criteria decision making* (MCDM) contribute to the analysis of decision-maker's preferences in a multidimensional criteria space where these criteria may either be quantifiable (for example, costs and physical distances) or non-quantifiable (for example, the aesthetical value of a layout).

Moreover, the criteria involved tend to be antagonistic: the improvement of one criterion can be accomplished only at the expense of another. As an illustration, suppose that two population centers at a certain distance from each other are to be served by a single hospital. Where should it be located? Formulated in terms of minimizing the sum of the distances travelled by the users, an optimal solution is to locate the hospital at the largest of the two centers. From an emergency point of view, however, the optimal solution is to take the midpoint between the two centers. Neither of these two solutions are optimal with respect to both criteria so which one, if any of the two, should actually be chosen by the decision-maker?

This or similar questions are inherent in MCDM and have no generally valid answers. The crux of the difficulty encountered is that the concept of optimality loses its significance for models involving two or more criteria. Suppose that a realistic decision problem is modelled in terms of minimizing k real-valued criterion functions $f_i(x)$, $i = 1, \dots, k$ subject to certain constraints. Ideally we seek a feasible solution x minimizing the *vector-valued function* $[f_1(x), \dots f_k(x)]$. Now, whereas an optimal solution to a single criterion optimization problem is unambiguously defined, there is no general definition of the minimum of a vector-valued function. Rather than optimality, a far more operational concept in this context is that of efficiency. For a pair (x, y) of feasible solutions we say that y *dominates* x if $f_i(y) \leq f_i(x)$, all i, and if strict inequality holds for at least one i. A feasible solution which is not dominated by any other feasible solution is called *efficient*.

The distinction between optimality and efficiency gives rise to a terminological schism: is "combinatorial MCDM" a subject within the field of combinatorial *optimization*? Since a COP, $\max\{ f(x) : x \in S \}$ can be viewed as a special case ($k = 1$) of the more general $\max\{ [f_1(x), \dots f_k(x)] : x \in S \}$, then indeed the converse question calls for an affirmative answer. Only tradition compels us to defend the opposite, and conceptually somewhat awkward position, namely that combinatorial programming with multiple criteria certainly is within the scope of combinatorial optimization.

2. THE SIGNIFICANCE OF COMBINATORIAL OPTIMIZATION

Disregarding combinatorial optimization as an attractive platform for academic careers, the three main justifications for the current and steadily increasing interest in the field are:

1. The variety of realistic decision problems amenable for modelling and analysis via combinatorial optimization.
2. The lack of a universal, operational algorithm.
3. Other theoretical challenges.

Like any other branch of applied OR, the ultimate goal of combinatorial optimization is (or should be) to offer decision support, that is, to provide decision-makers with a quantitative basis for finding good solutions to realistic decision problems.

To account for the postulated versatility of combinatorial optimization as a modelling tool, we shall here list selected application areas complemented by various model families. Based on our past as full-time consultants and supported by the experience gained by others, it is our contention that the most profitable application areas of combinatorial optimization include

- transportation, distribution
- sequencing, scheduling
- production planning
- locational analyses
- manpower planning
- investment

- network synthesis

These areas should certainly not be viewed as being disjoint; on the contrary, strong interrelations among two or more areas must often be taken into account when realistic problems are modelled. The word "profitable" does not (only) refer to the consultant's ability to make a decent living from his profession but relates in this context also to the decision-maker's expected gain. For example, locational analysis focus upon strategic rather than tactical matters, say, where to place factories or schools rather than how to steer the day-to-day deliveries from factories to retail outlets or how to route school buses. Such long-range decisions will normally involve more than just peanuts and substantial savings have often resulted from thorough investigations via appropriately designed optimization models.

Likewise, without dwelling on details, we hold the opinion that the most applicable and versatile model types or model families of combinatorial optimization include:

- travelling salesman
- set partitioning, set covering
- set packing
- knapsack
- network flows
- plant location, p-median, p-center
- quadratic assignment
- matching
- colouring
- spanning trees, Steiner trees

Most of the words listed above refer to specific, well-defined combinatorial optimization problems. At the same time, however, they can almost all be viewed as generic terms for extensive families of problems sharing certain common features. Consider, say, the so-called *simple plant location problem* (SPLP). While SPLP basically is a discrete, static, deterministic, one-product, fixed-plus-linear costs minimization problem, it can be modified to accommodate capacitated, dynamic, stochastic, multi-product, non-linear cost minimization formulations. It can moreover include prices as well as costs in its criteria and can also be used within the context of multicriteria decision-making. SPLP is thus not only a specific problem but should be regarded as the foremost member of a large family of locational decision problems.

Although e.g. traveling salesman is indispensable as the basic model for many realistic distribution and sequencing problems, there is not in general an obvious correspondence between the two lists. As an example, consider quadratic assignment, QAP. In addition to finding optimal layouts and/or analyzing alternative building designs, other fields of application of the QAP-model – far from the traditional domain of architects, building planners, and industrial engineers – have also been encountered. Backboard wiring problems, arrangement of electrical components in printed circuits, and arrangement of printed circuits themselves on a computer backplane are among the better known examples within electrical engineering. Other applications include planning of a presidential election campaign, arranging wedding guests around a table, scheduling parallel machines with change-over costs, and the design of typewriter keyboards. Personally, we have also met the problem in relation to finding the chronological order of 38 Babylonian texts and in setting up a model analyzing the movement of governmental institutions from Stockholm to a number of other Swedish cities.

The lack of a universal, operational algorithm.

From a practitioner's point of view, all of the normative models studied within combinatorial optimization are of interest only if operational algorithms exist for providing quantitative solutions to (non-contrived) data instances conforming to the model's structure. There is no rigorous definition of the term "operational"; for our purposes, however, "operational" implies empirically tested, known to possess provably true properties, and works well in practice.

For a linear programming (LP) problem with a bounded, feasible region defined by m linearly independent constraints expressed by m linear equations, and with n variables, the essence of the matter is, among the $\binom{n}{m}$ basic solutions, to find one which maximizes or minimizes a linear function. So defined, an LP-problem is clearly a COP. Irrespective of the specific pivot rules and updating schemes used, the *Simplex-type* (extreme point search) algorithms for general LP are all operational.

Actually, the same applies for an even larger class of COPs involving 0–1 variables, namely those sharing the property that an exact algorithm for the corresponding LP-relaxation is an exact algorithm for the original problem as well. This extended class of problems which accordingly is optimally solvable by any standard LP-code includes genuine combinatorial optimization problems like shortest paths, more sophisticated network flow problems, and matching.

The advent of the so-called ellipsoidal algorithms, and also the more recent Karmarkar algorithm and its descendants, which all, in contrast to the afore-mentioned Simplex-type algorithms, can be characterized as *interior point search* algorithms, and which all run in so-called polynomial time, once again stressed the computational tractability of general LP-problems. The same applies for the matching and the network flow problems referred to.

The claimed lack of a "universal operational algorithm" concerns virtually all other COPs of practical relevance. Cutting planes were introduced in the early 50's. For general integer linear programming (ILP), in principle encompassing all COPs, the cutting plane algorithms devised by Gomory by the end of the same decade do possess "provably true properties" but they are at the same time known *not* to perform well in practice. The concept of *strong valid inequalities* for COPs, however, have lead to new interest in cutting plane methods, since these in combination with search-based methods provide a very powerful tool for solving COPs.

Dynamic programming (DP), which also has influenced many other areas of optimization, experienced a great boom culminating in the late 50's. This technique, however, was later recognized as "fragile" in that the dimensions of the tables stored tended to be exceedingly large. DP was largely outperformed by an even more powerful principle for algorithmic design, known as *branch-and-bound*. Concepted in 1960 for general ILP by Land and Doig, and propelled into prominence in 1963 by Little et al., BB has still today maintained its status as the most versatile approach to solving combinatorial optimization problems. Literally all commercial codes for general ILP are nowadays based on BB with bounds generated via some LP-techniques.

Combining BB with cutting planes and strong valid inequalities lead in the late 1980s to the development of *branch-and-cut*, which is now the prime tool for solving COPs. If an acceptable ILP description of a COP can be given, the branch-and-cut method is superior to other methods not based on an LP-formulation of the problem. However, for some of the important COPs there is not much choice – either no ILP-formulation is known or the known ILP-formulations are not acceptable, for example due to an excessive number of variables. These problems are accordingly *not* solvable by branch-and-cut techniques. Instead, other efficient algorithms, often based on branch-and-bound and exploiting the special structure of the COP at hand, have been devised. This supports our claim that no "universal" algorithm exists.

Other challenges

Amidst the wealth of open questions of which some have been pending for years, we shall here mention a few challenges of interest to both theorists and practitioners concerned with combinatorial optimization.

For a given COP, $\min\{f(x) : x \in S\}$, an *exact algorithm* will produce a feasible solution x^E for which optimality is guaranteed, i.e. $f(x^E) = \min\{f(x) : x \in S\}$. A *heuristic* terminates with a feasible solution x^H which may or may not be optimal, that is, x^H is an *upper bound* on $f(x^H)$. If a bound on the deviation from the optimal solution value can be given, the heuristic is called an *approximation algorithm*. If some of the original constraints defining S are *relaxed* and if the *relaxed problem* is then solved to optimality, we obtain a solution x^R which normally is infeasible, and a value $f(x^R)$ of the objective function which constitutes a *lower bound* on $f(x^E)$.

For realistic decision problems amenable for modelling in terms of a COP, there is at times in practice a substantial gap between the dimensions of the problem at hand and the capabilities of algorithms available for its solution. Thus, proven optimality cannot always be hoped for in cases where no known exact algorithm is capable of handling a given data instance; we are then compelled to resort to an approximate algorithm or a heuristic. Approximation algorithms with tight bounds are known only for a limited number of COPs; hence, heuristics are by far the most common tool in finding feasible solutions to large data instances. Traditional heuristics are often based on pure common sense combined with practical experience. For example, if we were to find a minimum length tour visiting all the European capitals, why not just start somewhere and in each step continue to the nearest capital not yet visited before returning home? Realizing that the tour so found might intersect itself here and there, further improvement might be achieved by interchanging some of the cities.

For COPs in general, more refined heuristics may include algorithmic principles like *iterated descent* and *variable-depth local search*. *Tabu Search* is the generic name for heuristics based on local search equipped with memory. Furthermore, to avoid getting stuck in local minima and inspired by processes in physics and biology, a number of paradigms for heuristics using randomization have evolved during the last 10 years: *Simulated Annealing*, *Neural Networks*, and *Genetic Algorithms*.

The main idea of considering relaxations is to get rid of some "complicating constraints" and thereby to reduce a given COP to a simpler problem which then can be solved to optimality. As mentioned above, the result is in general an infeasible solution x^R and a lower bound $f(x^R)$ on $f(x^E)$. However, if we in addition have a feasible solution x^H generated by some heuristic and if the relative difference between the upper bound $f(x^H)$ and $f(x^R)$ is small, a good approach in practice is then to accept x^H as the ultimate solution to the given COP.

The significance of relaxations and heuristics can be summarized as follows: (1) no other options are at times available, (2) they are normally faster than exact algorithms by orders of magnitude, (3) via their capabilities of generating bounds, they are indispensable as components of exact algorithm, and (4) the performance of an exact algorithm can be considerably improved via a good starting solution generated by a heuristic.

The distinction between exact and approximate algorithms raises the fundamental question: how should approximate algorithms for solving a given COP be assessed?

Apart from empirical investigations, possibly guided by a decision-maker's opinion about the solutions proposed, there are basically two distinct approaches: worst-case analyses and probabilistic analyses. Upon defining an appropriate measure for a solution's deviation from optimality, one aim of *worst-case analyses* of approximate algorithms for a given problem type is to devise bounds for the maximum deviation and, if possible, to provide data instances to demonstrate that no better bounds exist. *Probabilistic analyses*, on the other hand, must be based on certain assumptions as to how the problem data are distributed. A measure of the "deviation" from optimality is then a random variable. Results provided via probabilistic analyses might be the probability that a data instance drawn at random satisfies some property, for example, that an approximate algorithm terminates with a solution within a prespecified percentage of optimality.

Worst-case analyses can also be conducted with an aim different from the one mentioned above, which concentrates solely on the value achieved of the objective function. We are here referring to worst-case analyses dealing with the so-called time complexity of a given algorithm designed for a specific problem and the family of all instances of a given size.

Probabilistic analyses, however, dealing as they do with algorithms' *average-case behaviour*, are at the same time more difficult and – at least from a practitioner's point of view – more pertinent than worst–case analyses irrespective of the goal they are pursuing. The most striking example is the Simplex–type algorithms for which highly contrived data instances have been constructed to demonstrate their exponential behavior in the worst case irrespective of the pivot rule employed. Nevertheless, such algorithms are recognized as working exceedingly well in practice. The opposite appears (so far) to apply for the family of ellipsoid algorithms which nevertheless have been proven to be polynomial time bounded. The status of the interior point algorithms in this respect is that it is known to outperform Simplex–type algorithms for a number of benchmark instances, but at the same time instances, for which the Simplex–type algorithms are superior, do exist.

Another direction for further research, though intimately related to the analyses of approximate algorithms, is that of computational complexity in general.

Via the definition of classes P and NP, followed by concepts like NP–completeness and NP–hardness, the theory of computational complexity has been instrumental in separating the relatively few COPs solvable by standard LP–codes from the plethora of problems which constitutes the real substance of combinatorial optimization. However, the fact that some NP–hard optimization problems in practice are computationally more demanding than others is still a constant source of dissatisfaction. While knapsack problems with tens of thousands of variables are easily solvable within reasonable time bounds, no exact algorithm is available for instances of quadratic assignment involving a few dozens of units to be placed. Practitioners have long ago developed their own lists of "easy" and "hard" NP–complete problems, and recent theoretical results seem to support this classification. The notion of *strong NP–completeness* classifying NP–complete problems according to their complexity if upper bounds are given on parameters representing values (as opposed to those serving as identifiers) is one of the advances. Knapsack is not strongly NP–complete as opposed to e.g. QAP and STSP. Also the extent to which a problem may be solved by a polynomial time approximation algorithm has given new insight into the classification of the NP–complete problems. Nevertheless, much further research in this direction is still called for.

Among "other challenges" we shall finally include the wealth of open questions penetrating the interfaces among combinatorial optimization and its neighbouring areas, notably graph theory and discrete mathematics in general. As evidenced by a steadily increasing number of publications, and the abundance of scientific meetings organized annually, it is not a too bold statement to say that a wonderful playground for academics has here emerged. More than twenty years after man first set foot on the moon, it is astonishing to see how certain, seemingly simple questions still today remain unanswered.

3. THE PAST AND THE PRESENT: A LITERARY DISCOURSE

The literature on combinatorial optimization has literally experienced an explosive growth over the past decades. To choose a few specimens for a short "literary discourse" from a market with high competition and with a wealth of high–quality products is difficult. To owe justice to all authors of recommendable books within our scope of interest is impossible.

The annotated bibliography provided below falls in two parts. First, a series of six classical "key" references from the 70's is presented. The second part comprises selected textbooks from the 80's and the 90's.

Six "key" references from the 70's

"Integer programming", "Combinatorial programming", "Discrete optimization", or "Combinatorial optimization"? These four variations of a theme are all represented in the titles of six books which all bear strong evidence as to the momentum gained by the field in the 70's.

One of the first, or possibly, *the* first meeting entitled "Combinatorial programming" was organized in Versailles in 1974. With the aim of becoming a handbook on the entire subject, the Proceedings were published a year later as:

- B. Roy (ed.), *Combinatorial Programming: Methods and Applications*, D. Reidel Publ. Co., Dordrecht, Boston, 1975.

It can safely be said that the opening sentence of the preface anticipated the future: " *"Combinatorial Programming" are two words whose juxtaposition still strike us as unusual, nevertheless their association in recent years adequately reflects the preoccupations underlying differing work fields, and their importance will increase both from methodology and application view points.*"

Another major event of that time was the workshop on Integer Programming held in Bonn in 1975 as documented in:

- P.L. Hammer, E.L. Johnson, B.H. Korte, and G.L. Nemhauser (eds.), "Studies in Integer Programming", *Annals of Discrete Mathematics* 1 (1977).

Here, the state of affairs is summarized in the preface as: "*There are a great many real-world problems of large dimension that urgently need to be solved but there is a large gap between the practical requirements and the theoretical development. Since combinatorial problems in general are among the most difficult in mathematics, a great deal of theoretical research is necessary before substantial advances in the practical solution of problems can be expected. Nevertheless the rapid progress of research in this field has produced mathematical results significant in their own right and has also borne substantial fruit for practical applications*".

In retrospect, however, and without detracting from the value of earlier endeavours, we find that the catalytic effect of two consecutive meetings held in Canada in 1977 can hardly be overrated. The organizers realized the appropriateness of assessing the current state of the entire subject and to examine its main trends of development. To this end, about 30 leading experts were commissioned one year in advance to prepare surveys on preassigned subfields within their main area of expertise. Most of the material was later made available in the two volumes:

- P.L. Hammer, E.L. Johnson, and B.H. Korte (eds.), "Discrete Optimization I and II ", *Annals of Discrete Mathematics* 4 and 5 (1979).

For the study of combinatorial optimization with particular emphasis on the complexity aspects, two books have occasionally been characterized as "indispensable". The first one is

- E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976

which focuses upon problems in P which can be formulated in terms of networks and algebraic structures known as matroids. The second work is the Lanchester Prize winning

- M.R. Garey, and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, 1979

which, in addition to the basic theory and directions for further study, provides an extensive list of NP-complete and NP-hard problems.

Selected textbooks from the 80's and the 90's

The following moderate-sized list of five books comprises three general textbooks plus two works devoted to specific application areas. The first textbook, which to some extent can be viewed as a synthesis of the above-mentioned works of Lawler, Garey and Johnson, is

- C.H. Papadimitriou, and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

The main objective is to integrate the computer scientists' ideas of computational complexity and the foundations of mathematical programming developed by the OR-community. This very readable volume does also contain a single chapter on the ellipsoid algorithm.

Another introduction to the entire field is

- G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1988

which deservedly is ranked as one of the best and most complete texts on the subject now available.

Educations towards the degree of Master of Business Administration are offered at several universities and business schools in Europe and have managed to attract post-graduate students from literally every corner of the world. Quantitative methods form a major ingredient in the curriculae. Among the subjects taught under this heading is model building in linear and combinatorial optimization with the main aim of advocating principles and recommending tools for investigating managerial and other decision problems via prescriptive models. Without familiarity with specific algorithms, the students are taught how to model relevant parts of the real world in such a way that the model is amenable for solution via some commercial software package. Thus, the students should become acquainted with the formats required; furthermore, they should appreciate the models' ability to answer "what if" questions which in practice is far more important than just providing the decision-maker with some "optimal" solution. MBA students as well as others interested in problem formulation and the art of model building will find an excellent guide in:

- H.P. Williams, *Model Building in Mathematical Programming*, John Wiley, 1993.

As was pointed out in the previous section, we listed routing and locational decisions among the most profitable application areas of combinatorial decisions. This viewpoint is supported by the titles of the following two, partially "applications-oriented" volumes.

As the foremost exponent of routing, the celebrated traveling salesman is the subject of:

- E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.), *The Traveling Salesman Problem*, Wiley, 1985.

This impressive work which apparently deals with a single model family only is not primarily addressing practitioners engaged in routing salesmen and the like. TSP owes rather its celebrity to its involvement in almost all of the major advances in combinatorial optimization as such. To mention a few highlights of the TSP-era in this respect: *Cutting planes* were originally proposed by Dantzig et al. (1954) as a technique for handling large-scale instances of TSP; the real potential of *branch-and-bound* as a general approach to combinatorial optimization problems became first apparent when Little et al. (1963) devised an exact algorithm for the symmetric TSP; within the same context, *Lagrangean relaxation* was put into use by Held and Karp (1970, 1971); two different versions of TSP were among the 21 decision problems whose *NP-completeness* is asserted in Karp's Main Theorem (1972); a cornerstone in the literature on *probabilistic analyses* of approximate algorithm was the investigations conducted by Karp (1977) with particular reference to TSP in the plane; finally Crowder and Padberg (1980) showed the potential of combining branch-and-bound and facet generation techniques in the solution of a 318-city TSP. Thus, the subtitle of this book, "A Guided Tour of Combinatorial Optimization", is well justified.

Since the contributors to the literature on locational analysis represent many diverse disciplines, such as general OR, graph theory and combinatorics, geography, regional science, and sociology, and since the relevant papers accordingly are scattered over many journals, there has for long been a strong need for up-to-date, all-round textbooks on the subject. The following book, however, fills part of the gap:

- R.L. Francis, and P.B. Mirchandani (eds.), *Discrete Location Theory*, Wiley, 1990.

Authored by 24 dedicated "locationists", this volume is mainly focussing upon the four most prominent families of models for analyzing locational decisions in discrete space: the uncapacitated plant location problem, the p-median and the p-center problem, and quadratic assignment, all quoted in the previous section. Both static and dynamic models are covered as are relatively new lines of research including congested networks, competitive location and spatial economics.

4. THE PROSPECTED FUTURE

"Predicting is difficult – especially predicting the future". This famous statement by the Danish humorist Storm Petersen is indeed true even if we restrict ourselves to the field of techniques and applications of combinatorial optimization. Nevertheless, we will in the following try to give a qualified guess, based partly on the actual development of combinatorial optimization, partly on similarities in the concurrent development of combinatorial optimization and computer science. The guess will mainly address combinatorial optimization methods as a modelling tool for decision makers whereas the prospected methodological/technical future is only mentioned briefly.

Combinatorial optimization (as we know it today) and computer science were both born around the late 40's and early 50's. A common characteristic for the two fields was that their establishment mainly was undertaken by those who really needed such tools for tackling realistic problems.

The following decade witnessed a rapid development. High-level programming languages as COBOL, FORTRAN, and ALGOL emerged, and computers grew more powerful by an order of magnitude. Standard techniques of optimization such as linear programming, network methods, cutting planes, and branch-and-bound were developed. For both subjects the applications became so diverse that the typical user of EDP/optimization techniques in the mid 60's no longer was an expert in either of the fields. The situation created a need for consulting companies with experts acting as interfaces between the decision-makers/EDP-users and systems.

From the mid 60's to the mid 70's the theoretical aspects of computer science and optimization still exhibited a lot of similarities. The tools were refined, and theories giving new insight as to what can and what cannot be accomplished were developed. In this context, computational complexity and computability deserve special mention. Also, the impact of heuristics as opposed to exact algorithms was recognized.

The consolidation of combinatorial optimization as a powerful tool for analyzing complex decision problems, however, did not change the by that time traditional relationship between an analyst and a decision-maker: upon having digested the problem posed by the decision-maker, the analyst created and solved the model "off-line". The results were then presented to the decision-maker for her approval.

New ideas as to the design of both hardware and software led to endless series of new applications of computer science. The advent of the PC was the final step so far in the process of making computers accessible to literally everyone. Obviously, it was no longer possible to maintain the traditional user/consultant scenario. A vast pressure on computer manufacturers and software houses to market systems usable even with limited knowledge of computer science followed in the wake and today's systems are visible results of these efforts.

In the last decade, optimization and computer science again show similarities with respect to speed of theoretical development. Of particular landmarks in optimization of special interest to combinatorial optimization, we note the polynomial algorithm for general LP devised by Khachian and more recently by Karmarkar, and the ingenious combination of branch–and–bound and facet generation techniques due to Johnson, Crowder and Padberg. Also the development within the field of average–case analysis of algorithms initiated by Karp should not be neglected. In computer science, the results on program generators and semantics of programming languages (denotational semantics) represent remarkable achievements as does the development of theory in the fields of distributed processes and parallel computation. Also the use of randomization as a tool in algorithmic design must be mentioned.

The boom in computer application seems to have created a need for, and inspired to research in new areas in a scale unparallelled in optimization. A few pertinent keywords include: user interfaces, 4th generation tools, expert systems, parallel architectures, and robotics.

Unfortunately, the catalyst effect of the PC on computer science has no obvious counterpart in optimization. No single factor has in a similar way accelerated the need for better means of handling questions of user communication, uncertainty, formulating and solving wrong problems, et cetera. By and large, the applied side of optimization is still facing these questions. In addition, the now customary use of PCs in other areas indicates that, unless such problems are seriously taken into account, optimization techniques may hardly survive as managerial tools. The positive side of the situation, however, is that we now, in contrast to the state–of–affairs in computer science 10 years ago, may draw upon the experiences gained in another field encompassed by the term: expert systems.

Expert systems are computer programs providing *expert–level* solutions to complex problems, usually by far too complex to model strictly mathematically. They are *heuristic* in that reasoning with judgmental knowledge as well as formal knowledge takes place; the are *transparent* in that explanations of the line of reasoning is provided, and they are *flexible* in that new knowledge is incrementally added to the existing body of knowledge of the system. To these very general statements may be added that expert systems often deal with uncertainty and that the knowledge of the systems is "general" as opposed to knowledge expressed by formulas, equations and inequalities.

Some of the points realized by expert system builders are in our opinion essential for the future success of optimization as a management tool: (1) Real–life problems are too complex to model using formal knowledge only, (2) solution processes are usually interactive, (3) to be trustworthy, problem solving systems must enable users to follow most of the solution process, and (4) even the best system will eventually fail (by not being used) if the user interface is poor.

To take advantage of the by now widespread familiarity with PCs among decision–makers, it is our belief that tomorrow's optimization systems should be designed with due regard to the following features:

- the systems must be *interactive*, thus enabling a solution process based on dialogues among the decision–maker, the analyst, and the computer.

- software for handling the mathematical optimization aspects of the problem at hand should constitute only a part (though probably an important one) of the entire system.
- devices for *integration* should be incorporated, i.e. in addition to specific solution techniques, the systems should provide an environment in which non-experts are assisted in formulating their problems. Also, the *presentation* of the results of the solution process is crucial to the applicability of the system. Note that *visual interactive modelling* managed to become an established term on the threshold to the 90' s.
- optimization systems are just another type of ordinary computer systems; hence all "rules of thumb" and all techniques developed in the fields of software engineering and systems analysis should be utilized wherever possible along with the construction of the system.

With respect to technical matters, optimization techniques have much to offer in the context of expert systems and logic programming systems based on search in a search space where complete enumeration is not possible. Combinatorial optimization has for decades been tackling this type of problems and the experience thereby gained should at least be made available to expert system builders. We also expect that considerable research efforts will be devoted to the by now somewhat controversial issue as to the potential impact of artificial neural networks in this respect.

As regards applications, it cannot be stressed too often that combinatorial optimization and computer science merely are tools to guide the decision process rather than providing decisive answers. It is all too easy to delegate responsibility for decision-making to a computerized system and hence to blame the model whenever undesirable effects result. For example, who was responsible for the "Black Monday", the stock jobbers or the computers?

We believe that the 90' s will be flooded with computerized decision support systems and that multicriteria models, in particular when (visual) interactive modes of operations are employed, are likely to play a far more dominant role for the design of such systems. Accordingly, the potential of these tools can only be fully recognized if they are used by interdisciplinary project groups capable of removing the traditional barriers among disciplines like social sciences, economics, engineering, et cetera. It is furthermore vital that decision-makers - be they private or public - take a holistic approach and concentrate on the *significant* problems of our societies.

There is little doubt that combinatorial optimization and computer science both can benefit from further cross fertilization. Rather than proceeding with new series of specialized meetings, journals, and curriculae, the communication lines among these and neighbouring fields should be further strengthened such that the different communities are kept aware of their neighbours' successes and failures.

We have not explicitly dealt with the postulated *beauty* of combinatorial optimization. It is not on, but only between the lines of this paper and the works referenced to therein. Likewise, we will leave it as postulate (which the reader may wish to verify for himself) that the beauty alone and the challenges that go with it are sufficient for combinatorial optimization to maintain its status as an academic discipline. It is nevertheless our hope that we in addition will witness the above-mentioned cross fertilization such that combinatorial optimization also remains recognized as an everyday tool for decision support in practice.