29 Extended Frontiers in Optimization Techniques

Sergiy Butenko and Panos M Pardalos

29.1 Recent Progress in Optimization Techniques

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound (Floudas and Pardalos 2002; Pardalos and Resende 2002). At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization today is a basic research tool in all areas of engineering, medicine and the sciences. The decision making tools based on optimization procedures are successfully applied in a wide range of practical problems arising in virtually any sphere of human activities, including biomedicine, energy management, aerospace research, telecommunications and finance. In this chapter we will briefly discuss the current developments and emerging challenges in optimization techniques and their applications.

The problems of finding the ``best'' and the ``worst'' have always been of a great interest. For example, given *n* sites, what is fastest way to visit all of them consecutively? In manufacturing, how should one cut plates of a material so that the waste is minimized? Some of the first optimization problems were solved in ancient Greece and are regarded among the most significant discoveries of that time. In the first century A.D., the Alexandrian mathematician Heron solved the problem of finding the shortest path between two points by way of the mirror. This result, also known as the Heron's theorem of the light ray, can be viewed as the origin of the theory of geometrical optics. The problem of finding extreme values gained a special importance in the seventeenth century when it served as one of motivations in the invention of differential calculus. The soon after developed calculus of variations and the theory of stationary values lie in the foundation of the modern mathematical theory of optimization.

The invention of the digital computer served as a powerful spur to the field of numerical optimization. During the World War II optimization algorithms were used to solve military logistics and operations problems. The military applications motivated the development of linear programming (LP), which studies optimization problems with linear objective function and constraints. In 1947 George Dantzig invented the simplex method for solving linear programs arising in U.S. Air Force operations. Linear programming has become one of the most popular and well studied optimization topics ever since. Although Dantzig is widely regarded as the father of linear programming, as early as 1939 the Soviet scientist Leonid Kantorovich emphasized the importance of certain classes of linear programs for applications in use of complex resources management, equipment work distribution, rational material cutting, the optimal use of sowing area, and transportation. He also proposed the method of resolving multipliers to solve these problems. Unfortunately, Kantorovich's work remained unnoticed until the linear programming methodology became a soundly developed and widely used discipline. But eventually the Kantorovich's contributions to the area of applied optimization were recognized by the Nobel Prize in Economics in 1975.

Despite of a fine performance of the simplex method on a wide variety of practical instances, it has an exponential worst-case time complexity and therefore is unacceptably slow in some large-scale cases. The question of existence of a theoretically efficient algorithm for LP remained open until 1979, when Leonid Khachian published his polynomial-time ellipsoid algorithm for linear programming. This theoretical breakthrough was followed by the interior point algorithm of Narendra Karmarkar published in 1984. Not only this algorithm has a polynomial-time theoretical complexity, it is also extremely efficient practically, allowing for solving larger instances of linear programs. Nowadays, various versions of interior point methods are an integral part of the state-of-the-art optimization solvers.

In nonlinear optimization, one deals with optimizing a nonlinear function over a feasible domain described by a set of, in general, nonlinear functions. The pioneering works on the gradient projection method by J. B. Rosen (Rosen 1960, 1961) generated a great deal of research enthusiasm in the area of for nonlinear programming, resulting in a number of new techniques for solving large-scale problems. This research resulted in several powerful nonlinear optimization software packages, including MINOS (Murtagh and Saunders 1983) and Lancelot (Conn et al. 1992).

In many practically important situations in linear, as well as nonlinear programming, all or a fraction of variables are restricted to be integer, yielding integer or mixed integer programming problems. These problems are in general computationally intractable, and it is unlikely that a universal polynomial-time algorithm will be developed for integer programming. Linear and integer programming can be considered as special cases of a broad optimization area called combinatorial optimization. In fact, most of combinatorial optimization problems can be formulated as integer programs. The most powerful integer programming solvers used by modern optimization packages such as CPLEX (ILOG 2001) and Xpress (Dash Optimization 2001) usually combine branch-and-bound algorithms with cutting plane methods, efficient preprocessing schemes, including fast heuristics, and sophisticated decomposition techniques.

It is fascinating to observe how naturally nonlinear and combinatorial optimization are bridged with each other to yield new, better optimization techniques. Combining the techniques for solving combinatorial problems with nonlinear optimization approaches is especially promising since it provides an alternative point of view and leads to new characterizations of the considered problems. These ideas also give a fresh insight into the complexity issues and frequently guide to discovery of remarkable connections between problems of seemingly different nature. For example, the ellipsoid and interior point methods for linear programming mentioned above are based on nonlinear optimization techniques. Let us also mention that an integrality constraint of the form x belongs to {0,1} is equivalent to the nonconvex quadratic constraint x^2-x = 0. This straightforward fact suggests that it is the presence of nonconvexity, not integrality that makes an optimization problem difficult (Horst et al. 2000).

Nonlinear programming techniques, in particular interior point methods, played a key role in the recent foundation and development of semidefinite programming. The remarkable result by Goemans and Williamson (Goemans and Williamson 1995) served as a major step forward in development of approximation algorithms and proved a special importance of semidefinite programming for combinatorial optimization.

The extensive field of global optimization copes with problems having multiple locally optimal solutions, which arise in various important applications. Since convexity of the objective function or the feasible region is difficult to verify for many problems, it makes sense to assume that these problems are multi-extreme and thus are of interest to the field of global optimization. Clearly, such problems are extremely difficult to solve, which explains the fact that only in the last few decades have solution techniques for global optimization problems been developed (Horst and Pardalos 1995). Recent advances in global optimization include efficient solution methods for nonconvex quadratic programming, general concave minimization, network optimization, Lipshchitz and DC (difference of convex) programming, multi-level and multi-objective optimization problems. More detail on these and other developments in global optimization can be found in (Horst and Pardalos 1995; Horst et al. 2000; Pardalos and Romeijn 2002).

In many optimization problems arising in supply chain management, resource allocation, inventory control, energy management, finance, and other applications, the input data, such as demand or cost, are stochastic. In addition to the difficulties encountered in deterministic optimization problems, the stochastic problems introduce the additional challenge of dealing with uncertainties. To handle such problems, one needs to utilize probabilistic methods alongside with optimization techniques. This led to the development of a new area called stochastic programming (Prekopa 1995), whose objective is to provide tools helping to design and control stochastic systems with the goal of optimizing their performance.

Due to a large size of most practical optimization problems, especially of the stochastic ones, the so-called decomposition methods were introduced. The decomposition techniques (Lasdon 1970) are used to subdivide a large-scale problem into subproblems of lower dimension which are easier to solve than the original problem. The optimal solution of the large problem is then found using the optimal solution of the subproblems. These techniques are usually applicable if the problem at hand has some special structural properties. Say, the Dantzig-Wolfe decomposition method (Dantzig and Wolfe 1960) applies to linear programs with block diagonal or block angular constraint matrices. Another popular technique used to solve large-scale linear programs of special structure is Benders decomposition (Benders 1962). One of the advantages of the decomposition approaches is that they can be easily parallelized and implemented in distributed computing environments.

The advances in parallel computing, including hardware, software, and algorithms, increase the limits of the sizes of problems that can be solved (Migdalas et al. 1997). In many cases, a parallel version of an algorithm allows for a reduction of the running time by several orders of magnitude compared to the sequential version. Recently, distributed computing environments were used to solve several extremely hard instances of some combinatorial optimization problems, for instance a 13,509-city instance of the traveling salesmen problem (Applegate et al. 1998) and an instance of the quadratic assignment problem of dimension 30 (Anstreicher et al. 2002). The increasing importance of parallel processing in optimization is reflected in the fact that modern commercial optimization software packages tend to incorporate parallelized versions of certain algorithms.

29.2 Heuristic Approaches

As a result of ongoing enhancement of the optimization methodology and of improvement of available computational facilities, the scale of the problems solvable to optimality is continuously rising. However, many large-scale optimization problems encountered in practice cannot be solved using traditional optimization techniques. A variety of new computational approaches, called heuristics, have been proposed for finding good sub-optimal solutions to difficult optimization problems. Etymologically, the word ``heuristic" comes from the Greek *heuriskein* (to find). Recall the famous ``Eureka, Eureka!" (I have found it! I have found it!) by Archimedes (287-212 B.C.).

A heuristic in optimization is any method that finds an ``acceptable'' feasible solution. Many classical heuristics are based on local search procedures, which iteratively move to a better solution (if such solution exists) in a neighborhood of the current solution. A procedure of this type usually terminates when the first local optimum is obtained. Randomization and restarting approaches used to overcome poor quality local solutions are often ineffective. More general strategies known as metaheuristics usually combine some heuristic approaches and direct them towards solutions of better quality than those found by local search heuristics. Heuristics and metaheuristics play a key role in the solution of large difficult applied optimization problems.

Sometimes in search for efficient heuristics people turn to nature, which seems to always find the best solutions. In the recent decades, new types of optimization algorithms have been developed and successfully tested, which essentially attempt to imitate certain natural processes. The natural phenomena observed in annealing processes, nervous systems and natural evolution were adopted by optimizers and led to design of the simulated annealing (Kirkpatrick et al. 1983), neural networks (Hopfield 1982) and evolutionary computation (Holland 1975) methods in the area of optimization. The ant colony optimization method presented in Chapter Five of this book is based on the behavior of natural ant colonies. Other popular metaheuristics include greedy randomized adaptive search procedures or GRASP (Feo and Resende 1995) and tabu search (Glover and Laguna 1997). Some of these and other heuristics and their applications in engineering were discussed in detail in previous chapters of this book. See also (Glover and Kochenberger 2003; Ribeiro and Hansen 2002).

29.2.1 Parallel Metaheuristics

Although metaheuristic approaches, in general, do not guarantee optimality, very often they offer the only practically feasible approach to solve large scale problems. However, with rapid growth of the scale of problems arising in science, engineering and industry, even metaheuristics may require computing times exceeding the limits of what is considered acceptable. *Parallel metaheuristics* are designed to deal with this and other problems encountered in heuristic approaches. In the remainder of this section we will give a brief review of developments and challenges in the area of parallel metaheuristics. More detailed surveys and further references can be found in (Correa et al. 2002; Glover and Kochenberger 2003; Migdalas et al. 1997).

The core idea of the parallel computing is to break up the assignment among several processors in order to accelerate the computation. In the case with metaheuristics, not only parallel computing helps to speed up finding good quality solutions, it also enhances the *robustness* of the approaches. The availability of multiple processors provides more opportunities for diversification of the search strategies thus widening the coverage of the solution space and yielding more reliable solutions. Improved robustness constitutes one of the most essential contributions of the parallel computing to metaheuristics.

The performance of a parallel algorithm depends on several key factors, such as the architectures of parallel machines used, the employed parallel programming environments, and the types of parallelism and models implemented.

The different architectures used in parallel computing include *shared-memory* machines and *distributed-memory* machines. The number of processors in modern shared memory multiprocessor machines (SMP) ranges from two to several hundred. One of the most popular currently used alternatives in parallel computing is represented by a *cluster of computers*, which is essentially a group of PCs or workstations connected through a network. The main advantage of the cluster systems is their good cost/performance ratios comparing to other parallel machines (Buyya 1999).

The basic choices of parallel computing environments consist of parallel programming languages, communication libraries, and programming with lightweight processes. Nowadays, there are many parallel computing tools available, such as Parallel Virtual Machine (PVM) (Geist et al. 1994), Message Passing Interface (MPI) (Gropp et al. 1998) and Linda (Carriero et al. 1994). In implementation of parallel metaheuristics, a proper programming tool can be selected depending on characteristics of a specific problem to be solved.

The two main types of parallelism are data parallelism and functional parallelism. In *data parallelism*, the same sequence of commands is carried out on subsets of the data, whereas in *functional parallelism* the program consists of cooperative tasks, which use different codes and can run asynchronously.

Apart from the different programming environments, there are two basic parallel programming models, namely centralized and distributed. The parallel implementations of metaheuristics are often based on hybrid models, in which a centralized model in shared memory multiprocessor machines is run under a distributed model used in the machines cluster.

Description of successful parallel implementations strategies of many various mataheuristics can be found in the literature. The metaheuristics that have been efficiently implemented in parallel environments include tabu search, GRASP, genetic algorithms, simulated annealing, ant colonies, and others. Recent surveys and references can be found in (Glover and Kochenberger 2003; Ribeiro and Hansen 2002).

29.3 Emerging Application Areas of Optimization

The fast pace of technological progress, the invention of Internet and wireless communications have changed the way people communicate and do business. This age of technology and electronic commerce creates new types of optimization problems. The enormous amounts of data generated in government, military, astronomy, finance, telecommunications, medicine, and other important applications pose new problems which require special interdisciplinary efforts and novel sophisticated techniques for their solution.

The problems brought by massive data sets include data storage, compression and visualization, information retrieval, nearest neighbor search, clustering, and pattern recognition among many others. These and other problems arising in massive data sets create enormous opportunities and challenges for representatives of many fields of science and engineering, including optimization. Some of these challenges are beginning to be addressed (Abello et al. 2002). For example, optimization algorithms on massive graphs have been recently applied to analyze the data modeled by these graphs (Boginski et al. 2003). There are still many questions to be formulated and answered in this extremely broad and significant area.

In many cases the data sets are too large to fit entirely inside the fast computer's internal memory, and a slower external memory (for example disks) needs to be used. The input/output communication (I/O) between these memories can result in an algorithm's slow performance. *External memory* (EM) algorithms and data structures are designed with aim to reduce the I/O cost by exploiting the locality. The first EM graph algorithm was developed by Ullman and Yannakakis in 1991 and dealt with the problem of transitive closure. Many other researchers contributed to the progress in this area ever since. Recently, external memory algorithms have been successfully applied for solving various problems, including finding connected components in graphs, topological sorting, and shortest paths. For more detail on external memory algorithms and data structures see (Abello and Vitter 1999; Vitter 2001).

Another area that has a huge potential for application of optimization techniques is biomedicine. In the last few years there have been successful attempts to employ optimization procedures in biomedical problems. For example, quadratic integer programming has been applied to find the optimal positioning of electrodes in the human brain used for detection and prediction of epileptic seizures (Iasemidis et al. 2001); network flow algorithms have been utilized in order to maximize efficiency and minimize risk in radiation therapy treatment; optimization has been used to improve the efficiency of cancer detection and treatment (Lee and Sofer 2003; Mangasarian et al. 1995). More references on applications of optimization techniques in biomedicine can be found in (Du et al. 2000, Pardalos and Principe 2002; Pardalos et al. 1996).

29.4 Concluding Remarks

The area of optimization is one in which recent developments have effected great changes in many other disciplines. This trend it seems will continue for the next several years. Taking into account the increasing demand for solving large scale optimization problems we are going to witness the development of powerful heuristics and their implementations in parallel computing environments in the near future. Although we singled out massive data sets and biomedicine as examples of extending applied optimization frontiers, many other important areas of human activities provide exciting opportunities and challenges for optimization techniques. In conclusion, we want to stress the greater than ever significance of optimization in solving interdisciplinary problems.

References

J. Abello, P.M. Pardalos, and M.G.C. Resende (editors). *Handbook of Massive Data Sets*. Kluwer Academic Publishers, 2002.

J. Abello and J. S. Vitter (editors). *External Memory Algorithms*. Vol. 50 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1999.

K.Anstreicher, N. Brixius, J.-P. Goux, and J. Linderoth. Solving large quadratic assignment problems on computational girds. *Mathematical Programming, Series B* 91 563-588, 2002.

D. Applegate, R. Bixby, V. Chv´atal, and W. Cook. On the solution of traveling salesman problems. *Doc. Math. J. DMV*, Extra ICM III:645–656, 1998.

J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4, 1962.

V. Boginski, S. Butenko, and P.M. Pardalos. Modeling and optimization in massive graphs. In P.M. Pardalos and H. Wolkowicz (editors), *Novel Approaches to Hard Discrete Optimization*. American Mathematical Society, 2003.

R. Buyya (editor). *High performance cluster computing: Architectures and systems* (in 2 volumes). Prentice Hall, 1999.

N. Carriero, D. Gelernter, and T. Mattson. The Linda alternative to message-passing systems. *Parallel Computing*, 20: 458-633, 1994.

A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *LANCELOT: a Fortran packagefor large-scale nonlinear optimization (Release A)*. Springer-Verlag, 1992.

R. Correa, I. Dutra, M. Fiallos, and F. Gomez (editors). *Models for Parallel and Distributed Computation: Theory, Algorithmic Techniques and Applications*. Kluwer Academic Publishers, 2002.

D.B. Dantzig and P. Wolfe. The decomposition principle for linear programs. *Operations Research*, 8, 1960.

Dash Optimization. Xpress, 2001. http://www.dashoptimization.com.

D.-Z. Du, P.M. Pardalos, and J. Wang (editors). *Discrete Mathematical Problems with Medical Applications*, DIMACS Series Vol. 55. American Mathematical Society, 2000.

T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 1995.

C. Floudas and P.M. Pardalos (editors). *Encyclopedia of Optimization (in 6 volumes)*. Kluwer Academic Publishers, 2002.

A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderman. *PVM: Parallel Virtual Machine – A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994. Also available at http://www.netlib.org/pvm3/book/pvm-book.html.

F. Glover and G.A. Kochenberger. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of ACM*, 42: 1115-1145, 1995.

W. Gropp, S. Huss-Lederman, A. Lumsdane, E. Lusk, B. Nitzberg, W. Saphir, and M. Snir. *MPI: the Complete Reference*. MIT Press, 1998. Also available at http://www.netlib.org/utk/papers/mpi-book/mpi-book.html.

J.H. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.

J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79: 2554-2558, 1982.

R. Horst and P.M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, 1995.

R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, 2nd edition, 2000.

L.D. Iasemidis, P.M. Pardalos, D.-S. Shiau, and J.C. Sackellares. Quadratic binary programming and dynamic system approach to determine the predictability of epileptic seizures. *Journal of Combinatorial Optimization*, 5: 9-26, 2001.

ILOG. ILOG CPLEX, 2001. http://www.ilog.com/products/cplex.

N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4: 373-395, 1984.

L.G. Khachian. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:1093–1096, 1979.

S. Kirkpatrick, C.D. Gellat Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

L.S. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970.

E. Lee and A. Sofer (editors). *Optimization in Medicine*. Volume 19 of *Annals of Operations Research*. Kluwer Academic Publishers, 2003.

O.L. Mangasarian, W.N. Street, and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43: 570-577, 1995.

A. Migdalas, P.M. Pardalos, and S. Storøy (editors). *Parallel Computing in Optimization*. Kluwer Academic Publishers, 1997.

B.A. Murtagh and M.A. Saunders. *MINOS 5.0 User's Guide*. Technical Report 83-20R, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1983.

P.M. Pardalos and J. Principe (editors). *Biocomputing*. Kluwer Academic Publishers, 2002.

P.M. Pardalos and M.G.C. Resende (editors). *Handbook of Applied Optimization*. Oxford University Press, 2002.

P.M. Pardalos and E. Romeijn. *Handbook of Global Optimization. Volume 2: Heuristic Approaches*. Kluwer Academic Publishers, 2002.

P.M. Pardalos, D. Shalloway, and G. Xue. *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, DIMACS Series Vol. 23. American Mathematical Society, 1996.

A. Prekopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

C. Ribeiro and P. Hansen (editors). *Essays and Surveys in Metaheuristics*. Kluwer Academic Publishers, 2002.

J.B. Rosen. The gradient projection method for nonlinear programming. Part I. Linear constraints. *J. Soc. Ind. Appl. Math.*, 8, 1960.

J.B. Rosen. The gradient projection method for nonlinear programming. Part II. Non-linear constraints. *J. Soc. Ind. Appl. Math.*, 9, 1961.

J. D. Ullman and M. Yannakakis. The input/output complexity of transitive closure. *Annals of Mathematics and Artificial Intelligence*, 3: 331-360, 1991.

J. S. Vitter. External memory algorithms and data structures: Dealing with MASSIVE DATA. *ACM Computing Surveys*, 33:209-271, 2001.