International Journal of Information Technology and Knowledge Management
January-June 2008, Volume 1, No. 1, pp. 147-154

GLOBAL OPTIMIZATION IN PRACTICE TODAY

Deepak Sharma, Parveen Sharma & Pankaj Sharma

Abstract

Meta-heuristics are renowned to present very efficient elucidation to many of today's combinatorial optimization problems in engineering, industrial, economical and scientific domains such as transportation, bioinformatics, logistics, business etc. Scheduling, timetabling, vehicle routing, resource allocation are intelligently and successfully tackled with Metaheuristic approaches such as Simulated Annealing, Tabu Search, Ant Colony Optimization, Harmony Search, Scatter Search, Iterated Local Search.

Metaheuristics present itself as highly promising choice for nearly-optimal solutions in reasonable time where exact approaches are not applicable due to extremly large running times or other limitations. Meta-heuristic is a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality.

This paper highlights the various contemporary real life applications of Metaheuristics in the domain of industrial engineering and NP-hard problems.

Keywords: Metaheuristics, Methods, Industrial and Engineering, Applications, Scope, Ant Colony Optimization

1. Introduction

Metaheuristics can solve Combinatorial Optimization Problems, like cutting and packing, routing, network design, assignment, scheduling, or time-tabling problems, continuous parameter optimization problems, or the optimization of non-linear strucutres like neural networks or tree structures as they often appear in computational intelligence.

Evolutionary Algorithms (EAs), in particular, comprise a variety of related algorithms that are based on the processes of evolution in nature. In contrast to several other Metaheuristics, they work on a set of concurrent solutions and can easily be parallelized.

Especially the combination of evolutionary algorithms with problem-specific heuristics, local-search based techniques, approximation methods and exact techniques often make possible highly efficient optimization algorithms for many areas of application.

Metaheuristics are generally applied to problems for which there is no satisfactory problem-specific algorithm or heuristic; or when it is not practical to implement such a method. Most commonly used Metaheuristics are targeted to combinatorial optimization problems, but of course can handle any problem that can be recast in that form, such as solving boolean equations.

In spite of overly-optimistic claims by some of their advocates, Metaheuristics are not a panacea, and their indiscriminate use often is much less efficient than even the crudest problem-specific heuristic, by several orders of magnitude.

Main Features of a Good Metaheuristics

- Population intrinsic parallelism
- Indirect Coding
- Cooperation adapted crossover
- Local search in solution space
- Diversity need to be controlled
- Easy to implement the restarts
- Randomness

Commonly used metaheuristic methods

- TS : Tabu search [Glover, 89 et 90]
- SA : Simulated annealing [Kirckpatrick, 83]
- TA : Threshold accepting [Deuck, Scheuer, 90]
- VNS : Variable neighborhood [Hansen, Mladenovi´c, 98]
- ILS : Iterated local search [Loren¸co et al, 2000]
- GA : Genetic Algorithm, Holland 1975 – Goldberg 1989
- MA : Memetic Algorithm, Moscatto 1989
- Hybrid Genetic Algorithm
- Ant Colony Optimization, Dorigo 1991
- Scatter search, Laguna, Glover, Marty 2000

Innumerable variants and hybrids of these techniques have been proposed, and many more applications of Metaheuristics to specific problems have been reported. This is an active field of research, with a considerable literature, a large community of researchers and users, and a wide range of applications.

2. ANT COLONY OPTIMIZATION

Ant Colony Optimization is one of the recognized technique in Metaheuristics for solving computational problems which can be reduced to finding good paths through graphs. They are inspired by the behaviour of ants in finding paths from the colony to food.

Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem. They have an advantage over simulated annealing and Genetic algorithm approaches when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

Current applications of ACO algorithms fall into the two important problem classes of static and dynamic combinatorial optimization problems. Static problems are those whose topology and cost do not change while the problems are being solved. This is the case, for example, for the classic TSP, in which city locations and intercity distances do not change during the algorithm's run-time. Differently, in dynamic problems the topology and costs can change while solutions are built.

An example of such a problem is routing in telecommunications networks, in which traffic patterns change all the time.

3. REAL LIFE INDUSTRIAL APPLICATIONS OF ANT COLONY OPTMIZATION

Ant Colony Optimization efficiently Solves NP hard Problems. One of the best examples of an NP-hard problem is the decision problem **SUBSET-SUM** which is described as : given a set of integers, does any non empty subset of them add up to zero? It means a *yes*/*no* question, and happens to be NP-complete.

We can also elaborate it as : Given a set of integers, does the sum of some non-empty subset equal exactly zero? For example, given the set { "7, "3, "2, 5, 8}, the answer is YES because the subset { "3, "2, 5} sums to zero.

4. FAMOUS REAL LIFE ENGINEERING AND INDUSTRIAL APPLICATIONS

- Routing
  - TSP (Traveling Salesman Problem)
  - Vehicle Routing
  - Sequential Ordering
- Assignment
  - QAP (Quadratic Assignment Problem)
  - Graph Coloring
  - Generalized Assignment

  - Frequency Assignment
  - University Course Time Scheduling
- Scheduling
  - Job Shop
  - Open Shop
  - Flow Shop
  - Total tardiness (weighted/non-weighted)
  - Project Scheduling
  - Group Shop

Large number of different ACO algorithms to exploit different problem characteristics

Ant Colony Optimization

Generate an initial colony
While stopping conditions are not met
For each ant of the colonyinitialize ant memory
While Curent State = Target state
Apply ant decision under pheromone information
Move to next state
Deposit pheromone information on the transition
End While
End For
Evaporate pheromone information
End While

Solving SUDOKU Puzzles : Metaheuristics in Game Theory

[Rhyd Lewis, Centre for Emergent Computing, Napier University, Scotland]

It was the first application of a metaheuristic technique to the popular sudoku puzzle. When applying this algorithm to a large number of problem instances taken from the UK press (of various degrees of 'difficulty'). The algorithm does not just get close to optimality (as is often the case with optimization techniques), but consistently finds the solution in reasonably short amounts of time. It was proved that this method, particularly for lower order puzzles, is more robust than many existing algorithms, because in order to be successful, it does not necessarily depend on problem instances being logic-solvable.

Through a large number of experiments with the instance generator, the study witnessed the existence of easy-hard-easy style phase-transition in larger puzzles, which are similar to those found in other NP-complete problems such as constraint satisfaction problems (Smith, 1994), timetabling problems (Ross, Corne, and Terashima-Marin, 1996), and graph colouring problems (Turner, 1988).

Other techniques used for solving NP-Hard and other Computational Problems and optimization are:

Genetic Algorithms (GA)

Genetic Algorithms (GA) maintains a pool of solutions rather than just one. The process of finding superior solutions mimics that of evolution, with solutions being combined or mutated to alter the pool of solutions, with solutions of inferior quality being discarded.

Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Simulated Annealing (SA)

Simulated Annealing (SA) is a related global optimization technique which traverses the search space by generating neighboring solutions of the current solution. A superior neighbour is always accepted. An inferior neighbour is accepted probabilistically based on the difference in quality and a temperature parameter. The temperature parameter is modified as the algorithm progresses to alter the nature of the search.

In the simulated annealing (SA) method, each point s of the search space is compared to a state of some physical system, and the function E(s) to be minimized is interpreted as the internal energy of the system in that state. Therefore the goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy.

Tabu Search (TS)

Tabu search (TS) is similar to Simulated Annealing, in that both traverse the solution space by testing mutations of an individual solution. While simulated annealing generates only one mutated solution, Tabu search generates many mutated solutions and moves to the solution with the lowest fitness of those generated. In order to prevent cycling and encourage greater movement through the solution space, a Tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the Tabu list, which is updated as the solution traverses the solution space.

Tabu Search is a powerful algorithmic approach that has been applied with great success to many difficult combinatorial problems. A particularly nice feature of TS is that, like all approaches based on Local Search, it can quite easily handle the "dirty" complicating constraints that are typically found in real-life applications. It is thus a really practical approach. It is not, however, a panacea: every reviewer or editor of a scientific journal has seen more than his/her share of failed TS heuristics.

These failures stem from two major causes: an insufficient understanding of fundamental concepts of the method (and we hope that this tutorial will help in alleviating this shortcoming), but also, more often than not, a crippling lack of understanding of the problem at hand. One cannot develop a good TS heuristic for a problem that he/she does not know well! This is because significant problem knowledge is absolutely required to perform the most basic steps of the development of any TS procedure, namely the choice of a search space and of an effective neighborhood structure. If the search space and/or the neighborhood structure are inadequate, no amount of TS expertise will be sufficient to save the day. A last word of caution: to be successful, all meta-heuristics need to achieve both depth and breadth in their searching process; depth is usually not a problem for TS, which is quite aggressive in this respect (TS heuristics generally find pretty good solutions very early in the search), but breadth can be a critical issue. To handle this, it is extremely important to develop an effective diversification scheme.

Conclusion

However, lot of advancements has been pursued in finding exact solutions to the combinatorial optimization problems using techniques such as integer programming, dynamic programming, cutting planes, and branch and cut methods, still there are many hard combinatorial problems which are unsolved and require good heuristic methods. "Optimal Solutions" is in many cases meaningless, as in practice we are often dealing with models that are rough simplifications of reality. The goal of Metaheuristics is to produce good-quality efficient solutions without necessarily providing any guarantee of solution quality. Modern Metaheuristics include Simulated Annealing, Genetic Algorithms, Tabu Search, GRASP, ant colony optimization, and their hybrids. No doubt Metaheuristics have been one of the most stimulating topics to explore in the field of optimization.

REFERENCES

[1] Fleurent C. and J. A. Ferland, (1996), "Genetic and Hybrid Algorithms for Graph Colouring," *Annals of Operations Research*, **63**, 437-461.

[2] Gendreau M., (2002), "Recent Advances in Tabu Search," In Essays and Surveys in Metaheuristics, C. C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, 369-377.

[3] Hertz A., and D. de Werra (1991), "The Tabu Search Metaheuristic: How We Used It," *Annals of Mathematics and Artificial Intelligence*, **1**, 111-121.

[4] Kirkpatrick S., C. D. Gelatt Jr. and M. P. Vecchi, (1983), "Optimization by Simulated Annealing," *Science*, **220**, 671-680.

[5] Laporte G. and I. H. Osman (eds.), (1996), "Metaheuristics in Combinatorial Optimization," *Annals of Operations Research*, **63**, J.C. Baltzer Science Publishers, Basel, Switzerland.

[6] Lokketangen A. and F. Glover, (1996), "Probabilistic Move Selection in Tabu Search for 0/1 Mixed Integer Programming Problems," In Meta-Heuristics: Theory and Applications, I. H. Osman and J. P. Kelly (eds.), Kluwer Academic Publishers, 467-488.

[7] Osman I. H., (1993), "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem," *Annals of Operations Research*, **41**, 421-451.

[8] Osman I. H. and J. P. Kelly, (eds.), (1996), Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, Norwell, MA.

[9] Pesant G. and M. Gendreau, (1999), "A Constraint Programming Framework for Local Search Methods," *Journal of Heuristics*, **5**, 255-280.

[10] Ribeiro C. C. and P. Hansen, (eds.), (2002), Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, Norwell, MA.

[11] Camp Charles V., Bichon Barron J. and Stovall Scott P., (2005), "Design of Steel Frames Using Ant Colony Optimization," *Journal of Structural Engineeering*, **131**(3), 369-379.

[12] Fjalldal Johann Bragi, (2005), "An Introduction to Ant Colony Algorithms," *http://www.informatics.sussex.ac.uk/research/nlp/gazdar/teach/atc/1999/web/johannf/ants.html*, Accessed.

[13] Geem Z. W., (2006), "Optimal Cost Design of Water Distribution Networks using Harmony Search," *Engineering Optimization*, **38**(3), 259-280.

[14] Geem Z. W., (2006), "Comparison Harmony Search with Other Meta-Heuristics in Water Distribution Network Design," Proceedings of *8th Annual International Symposium on Water Distribution Systems Analysis (WDSA 2006)*, ASCE, CD-ROM, Cincinnati, OH, USA,.

[15] Lee, K. S. and Geem Z. W., (2004), "A New Structural Optimization Method Based on the Harmony Search Algorithm," Computers & Structures, **82**(9-10), 781-798.

[16] Erdal F., and Saka M. P., "Optimum Design of Grillage Systems Using Harmony Search.

[17] Algorithm," Proceedings of 8th International Conference on Computational Structures.

[18] Technology (CST 2006), (2006), Las Palmas de Gran Canaria, Spain.

[19] Geem Z. W., Lee K. S., and Tseng C. -L., "Harmony Search for Structural Design."

[20] Geem Z. W., Lee K. S., and Park Y., (2005), "Application of Harmony Search to Vehicle Routing," *American Journal of Applied Sciences*, **2**(12), 1552-1557.

Deepak Sharma
CSE Department, GIMT, Kanipla, Kurukshetra University,
Kurukshetra, Haryana, INDIA
E-mail: sharmadeepak2k4@gmail.com

Parveen Sharma
CSE Department, SKIET, Kurukshetra, Kurukshetra University,
Kurukshetra, Haryana, INDIA
E-mail: sharmaparveen19@gmail.com

Pankaj Sharma
Department of Computer Sc. & Applications, K.U., Kurukshetra,
Haryana, INDIA
E-mail: pankaj2004gla@gmail.com