

**SOCIAL NETWORK OPTIMIZATION  
A NEW METHAHEURISTIC FOR GENERAL OPTIMIZATION PROBLEMS**

**OTIMIZAÇÃO USANDO REDE SOCIAL  
UMA NOVA METAHEURÍSTICA PARA PROBLEMAS GERAIS DE OTIMIZAÇÃO**

Hassan Sherafat<sup>1</sup>

<sup>1</sup>Universidade Federal de Sergipe – UFS – São Cristóvão/SE – Brasil  
[hassan@ufs.br](mailto:hassan@ufs.br)

**Abstract**

*In the recent years metaheuristics were studied and developed as powerful technics for hard optimization problems. Some of well-known technics in this field are: Genetic Algorithms, Tabu Search, Simulated Annealing, Ant Colony Optimization, and Swarm Intelligence, which are applied successfully to many complex optimization problems. In this paper, we introduce a new metaheuristic for solving such problems based on social networks concept, named as Social Network Optimization – SNO. We show that a wide range of np-hard optimization problems may be solved by SNO.*

**Key-words:** Metaheuristic, Computation, Optimization, np-hard problems.

**Resumo**

*Nos últimos anos, as meta-heurísticas foram estudadas e desenvolvidas como técnicas poderosas para solução de problemas de otimização de difícil tratamento. Algumas das técnicas conhecidas neste campo são: Algoritmos Genéticos, Pesquisa Tabu, Recozimento Simulado, Otimização por Colônia de Formigas e Inteligência de Enxames, as quais são aplicadas com sucesso em muitos problemas complexos de otimização. Neste artigo, apresentamos uma nova meta-heurística para resolver tais problemas com base no conceito de redes sociais, denominada Otimização usando Redes Sociais - SNO. Mostraremos que uma grande variedade de problemas de otimização do tipo NP-hard podem ser resolvidos pela SNO.*

**Palavras-chave:** Meta-heurística, Computação, Otimização, problemas NP-hard.

## 1. Scope

In applied mathematics and computation some problems are regarded as np-complete, or np-hard problems (GAREY, 1979). There are evidences that any solution for these problems involves an exponential number of operation, which makes it impossible to solve the problem for large, even medium sized problems. A classical example of a np-complete problem is the well-known Travelling Salesman Problem -TSP, in which  $n$  cities should be visited in a circuit, where the distance (cost) between each pair of cities is known. The problem is: in which sequence the cities should be visited in order to minimize the total distance travelled. Any exact solution for TSP leads to an implicit enumeration of possible circuits. And we know that the total number of possible circuits are  $(n-1)!$  Just to have an idea, a TSP with 60 cities may have up to  $1.38 \cdot 10^{80}$  Possible solutions. Other problems in the class np-complete have the same behavior. This is the reason for employing and searching approximate solutions for such problems.

Approximate solutions may be provided by heuristic methods, designed for each case. There exist several approaches for heuristics: greedy methods, local searches, and many others. The main problem for these alternatives is the worst case: heuristics may easily get trapped in a local optimum, and the approximate solution be far from the global optimum.

In counterpart, metaheuristics are sophisticated methods that employ a simple heuristic which exploit intelligently the whole solution space, without fear of falling into a local optimum. The first time that the term metaheuristic was used was by Glover (1986). Some well-known metaheuristics which have been successfully applied to combinatorial optimization problems are the following:

**Tabu search**, created by Glover (1986), is a metaheuristic employing local search methods. The local (neighborhood) search takes a potential solution to a problem and check its immediate neighbors (that is, solutions that are similar except for one or two minor details) in the hope of finding an improved solution. Local search methods have a tendency to become stuck in suboptimal regions or on plateaus where many solutions are equally fit.

Tabu search improves the performance of local search by relaxing its basic rule. First, at each step *worsening* moves can be accepted if no improving move is available (like when the search is stuck at a strict local optimum). In addition, *prohibitions* (henceforth the term *tabu*) are introduced to discourage the search from coming back to previously-visited solutions.

**Genetic Algorithm**, suggested by Holland (1975), is a metaheuristic inspired in natural selection by relying on two bio-inspired operators: crossover and mutation. In a genetic algorithm, a population of candidate solutions (called individuals) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties which can be mutated and altered.

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, called the initial *generation*. In subsequent generations, the fitness of every individual in the population is evaluated; the fitness is usually calculated from the value of the objective function in the problem being solved. The more fit individuals are stochastically selected from the current population, and new ones are created using two basic operations: crossover and mutations. The new generation of candidate solutions 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.

**Simulated annealing**, developed by Kirkpatrick et al. (1983) is a probabilistic technique for approximating the global optimum of a given function. The name and inspiration come from annealing in metallurgy. Simulated annealing interprets slow cooling as a slow decrease in the probability of accepting worse solutions as it explores the solution space.

Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the optimal solution. This notion of slow cooling is implemented in the Simulated Annealing algorithm as a slow decrease in the probability of accepting worse solutions as it explores the solution space.

**Ant Colony Optimization**, devised by Dorigo (1992) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. The technique may be seen as a member of family of Swarm Intelligence methods.

the first algorithm of Dorigo was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.

Metaheuristics cited above have inspired several other techniques, studied in literature as Evolutionary methods, Memetic algorithms, and the Swarm Intelligence (BIANCHI et al., 2009).

## 2. Introduction

In this paper, we propose a new metaheuristic named Social Network Optimization – SNO, applied to general optimization problems. The method is inspired in learning and interacting of individuals in a social network, where each member is in contact with a limited number of other members (friends). And his friends, in turn, contact a finite number of other friends, who may be, or not common friends with the first one, or each with other.

Each individual in SNO represents a solution for the optimization problem. Then one solution may be compared with those solutions in the network which it has contact. The idea is that the individual may learn a detail from another individual in his friendship circle. Other members of the Network practice the learning in same fashion. In one iteration of the algorithm all individuals try to update their solutions having good influences from his network.

As in real social networks, the interaction between individuals can lead to collective learning. Learning would become more intense and meaningful if there is a learning posture for the members. This is the basic idea of SNO, since learning is provoked. In each iteration, all individuals have a chance to learn a little more, in some aspect of the solution, taking advantage of the natural diversity that exists in the network.

### 3. Social Network Concept

A graph  $G = (X, E)$  can represent a social network, where each vertex  $i \in X$  represents an individual and each edge  $(i, j) \in E$  the friendship between individuals  $i$  and  $j$ . Let  $n = |X|$  be the number of vertices and  $m = |E|$  the number of edges of the graph. For practical purposes, at the time of implementation of the algorithm, it would be better that the number of contacts for each individual be a constant  $d$ , that is, each one has exactly  $d$  friends. That means, the degree of every vertex in corresponding graph is  $d$ . This assumption is possible, provided that  $d < n$  and at least one of these numbers ( $n, d$ ) is even. The last condition must be true, because the sum of all degrees of vertices is twice the number of edges:

$$nd = 2m$$

Therefore  $m = nd/2$ , that implies  $n$  or  $d$ , one of them should be even.

The numbers  $n$  and  $d$  are two important parameters of the algorithm and should be properly fixed. Higher the number of variables of the problem, larger should be the number  $n$ . While  $d$  limits the domain of local search in the learning operation. For a fast algorithm,  $d \ll n$ .

The graph  $G$  is randomly created. In its construction, should be observed its connectivity, since a disconnected graph (with two or more disjoint components) implies the impossibility of direct or indirect learning between disconnected individuals.

To each vertex of the graph is associated a solution for the problem being solved. The initial solution may be random, or generated using some heuristic, for example a greedy stochastic method. In this case, it must be ensured that the heuristic does not generate very similar solutions. The most important quality for an initial solution is its diversity.

#### 4. Learning Operation

SNO is an evolutionary method, in which the evolution of individuals is through learning, than by means of blind combinations, or simple imitation of a majority. In genetic algorithms, for example, the improvement from one generation to another is achieved from recombination of individuals, privileging those who have good adaptation. In Ant Colony algorithm, the improvement occurs by means of a probabilistic imitation of the majority of other ants. Although these procedures could be interpreted as forms of learning, however, they cannot be considered as conscious learning.

An important feature of SNO is its gradual evolution through learning: a conscious learning, either from a good individual, or from a very poor one. Learning does not mean exchanging an entire solution by another. It means to assimilate a detail of another solution that gives you some advantage. The operation that leads to learning is through some comparative local search between an existing solution at a vertex and the solutions of its friends. Such operation may be performed using several strategies, depending on the nature of the problem being solved. We present below a possible generic strategy:

Let  $S^k(i)$  be the current solution (in iteration  $k$ ) of an individual  $i$ , with objective function value  $f_i^k$ . Suppose  $\sigma$  is a detail of this solution, which appears as  $\sigma_i$  in  $i$ . Let  $J = \Gamma(i)$  be the set of friends of  $i$  in the network. The detail  $\sigma$  appears as  $\sigma_j$  in each  $j \in J$ . If individual  $i$  assimilate  $\sigma_j$  in his solution, then his objective function value will be changed to  $f_i^k(\sigma_j)$ . Logically  $i$  would accept the change from a  $j$  who gives him the best advance, and only if this is better than his actual solution.

However, the new solution  $S^{k+1}(i)$  should take into account two parameters: the value of the objective function  $f_j^k$  (from whom is getting information) and the improvement caused in his own objective function value  $f_i^k(\sigma_j)$ , assimilating  $\sigma_j$ . In other words, it is important the improvement achieved in the value of the objective function, and equally important the objective function value of whom contributed with development.

The importance of each of these components should be appropriately scaled up, using parameters  $\alpha, \beta \geq 0$ , and for a minimization problem assessing

$$\varphi_i = \max_j \alpha [f_i^k - f_i^k(\sigma_j)] + \beta [f_i^k - f_j^k].$$

Then the new solution for  $i$  will be  $S^{k+1}(i)$ , obtained from learning of solution detail  $\sigma_j$  from individual  $j$ , who maximizes the value of  $\varphi_i$ . If  $\varphi_i \leq 0$ , then  $S^{k+1}(i) = S^k(i)$ .

Note the learning process does not affect no individuals in  $J$ , which remain intact.

## 5. The SNO Algorithm

A typical algorithm for SNO may be as follows:

0) Given a minimization problem  $P$ , with following parameters:

$n$ : total number of members in the social network (even number);

$d$ : number of friends for each member;

$\alpha, \beta \geq 0$ : learning parameters.

1) Create a random graph  $G = (X, E)$  with  $n$  vertices, all of them with degree  $d$ , representing the social network.

2) Associate to each vertex  $i \in X$  a solution  $S^0(i)$ , obtained by some greedy stochastic method, and calculate the solution value  $f_i^0$ .

Fix  $k = 0$ .

3) While some convergence condition is not satisfied, repeat steps 3.1 through 3.2:

3.1. Repeat steps 3.1.1 through 3.1.2, while exist individuals who has not learned in iteration  $k$ ,

3.1.1. Select an arbitrary individual  $i \in X$  who has not yet learned in the present iteration;

3.1.2. Individual  $i$  learn a solution detail  $\sigma$  from his friends  $j \in \Gamma(i)$ , by maximizing  $\varphi_i$ :

$$\varphi_i = \max_j \alpha [f_i^k - f_i^k(\sigma_j)] + \beta [f_i^k - f_j^k].$$

- If  $\varphi_i > 0$ , then improve the solution  $S^{k+1}(i)$ , by assimilating  $\sigma_j$ , which maximizes  $\varphi_i$ , as mentioned above.
- If  $\varphi_i \leq 0$ , make  $S^{k+1}(i) = S^k(i)$ .

3.2.  $k = k + 1$ .

4)  $S^k(i)$  for which  $f_i^k = \min_i \{f_i^k\}$  is the best solution obtained for problem  $P$ . //

## 6. Convergence

A possible stopping criterion for the algorithm may be the non-learning condition in an entire iteration. That is, no individual could improve his solution through learning. What means such a condition? It may be a local, or global optimum.

The convergence of SNO to the global optimum solution, in a finite number of iterations, cannot be guaranteed, nor for any other metaheuristic. What may be established is that the method has ability to leave local optimums. The learning process is unilateral: individual  $i$  learns from  $j$ , but do not teach him! Instead,  $j$  learns from  $k$ , and so on. Furthermore, the solution detail (object of

learning) may randomly be changed. The result is a population that evolves but maintains heterogeneity. And this is a favorable condition for getting out from a local optimum.

## 7. Additional Strategies

What we presented as an algorithm is only an option for a basic structure of SNO. The success of the algorithm depends greatly on the strategies adopted in its implementation for the specific problem being solved. Additionally, some other strategies could be considered:

The network construction may be different, and its vertices with varying degrees; In the construction of the network, the degree of the vertices need not to be necessarily a constant. That means, the number of friends for each individual can be different.

After the culmination of the evolution process, the same individuals (already evolved) may change their friendship network, continuing the process with the new network.

The learning strategy may take several forms. For example, if an individual does not improve his solution by learning a detail, the detail can be changed.

A **Conference** strategy may be implemented in the algorithm after performing a predefined number of iterations. The conference aims for an additional learning between a number of high standing individuals, no matter if they are, or not, in a friendship circle.

## 8. Conclusions

We present a new metaheuristic, named Social Network Optimization – SNO, inspired on social networks behavior, to solve general optimization problems. The SNO is based on a systematic learning of all individuals, each one from others in own friendship network. We showed capability of SNO to overcome local optimums toward better solutions, however, not affirming that it achieves the best one. The methodology was presented in conceptual form, applicable to a variety of problems of difficult treatment.

## References

- BIANCHI, L.; DORIGO, M.; GAMBARDELLA, L.M.; GUTTJAHN, W.J. (2009). "A survey on metaheuristics for stochastic combinatorial optimization". *Natural Computing: an international journal*. **8** (2): 239–287.
- HOLLAND, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Papadimitriou C. H.
- KIRKPATRICK, S.; GELATT Jr., C.D.; VECCHI, M.P. (1983). "Optimization by Simulated Annealing". *Science*. **220** (4598).

GAREY, M. and JOHNSON, D., *Computers and Intractability; A Guide to the Theory of NP-Completeness*, 1979..

GLOVER, F. (1986). "Future Paths for Integer Programming and Links to Artificial Intelligence". *Computers and Operations Research*. **13** (5): 533–549.

DORIGO, M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italie, 1992.

VOUDORIS C., TSANG E., Guided Local Search and its Application to the Travelling Salesman Problem, *European Journal of Operations Research* 113, pag. 469-499, 1999.

Recebido: 05/03/2017

Aprovado: 15/11/2017