METAHEURISTIC HYBRIDIZATION: MEMETIC ALGORITHM

Stefan Andreea-Mirabela, PhD Student
University of Craiova
Faculty of Economics and Business Administration
Craiova, Romania

Abstract: In today's times, the majority of investors are willing easy gains, and the easiest way is to invest in productive domains, even if the risks they have to take are very high, but these are not the last problem. The real problem for investors is to take the right decision and, for that, they have to build forecast models, to make some assumptions for default variables, which are difficult to understand. To refine the prediction process, researchers have developed hybrid models for forecasting. A famous class of all these techniques is in the metaheuristics domain. The purpose of this paper is to present the hybridization process and hybrid algorithms. More over, another goal is to reveal some possibilities of metaheuristics, such as hybridization of algorithms.

JEL classification: C10, C18

Key words: hybrid metaheuristics, heuristics, genetic algorithms, metaheuristics, memetic algorithm.

1. INTRODUCTION

Hybridization is a trend remarked in numerous researches on metaheuristics over the past years. This means taking advantage of the remarkable advantages of different metaheuristics, given that many metaheuristics, known so far, still remain in force, until the time of research and improvement, to find a quick solution to optimization problems. Hybridization consists of mixing the features of two distinguished methods to result the advantages of the two combined methods. The begenings of hybrid metaheuristic algorithms start to the work of Glover. He gain a simple descent method to improve evolutionary research. But in those days, most researchers paid little attention to it. Nowadays, hybrid metaheuristics have become more popular because the best results found for several combinatorial optimization problems have been obtained with hybrid algorithms.

In recent years, the hybridization of algorithms has attracted the attention of many researchers in order to improve their performance. The hybridization goal is to mix the characteristics of several algorithms to profit of their advantages. But the obtaining algorithm may also have their defects. More over, an hybridization resulted algorithm from the of several algorithms can have significant comprehensivity.

Also, genetic algorithms require a big evaluation number of the performance function (large number of iterations and a large population) for the purpose to obtain good results. So, for comprehensive combinatorial problems, genetic algorithms are being harmed by the complexity of the calculations. Also, the idea came from using hybrid algorithms which benefict of the complementarity of more methods between them. Hybrid algorithms rarely combine evolving metaheuristics and a local search method. So by using this, the power of the evolutionary method makes it possible to sweep solutions in a global way. The local search method focuses on exploring a small area of this space in order to perfect the solution. This team work can take the form of a simple handover between the methods used. It is a simple form of hybridization. But the two approaches can also be intertwined in a more comprehensive way. More over, by using an example for hybridizing two algorithms X and Y, is obtaining a new complex algorithm which is no longer totally X or Y, but rather something which is a combination of the two algorithms. So, memetic algorithms are computerized intelligence structures that mix various operators to find a solution to optimization problems. The complexity of operator selection underpins memetic algorithms and their ability to solve difficult problems. The importance of memetic algorithms lies in the fact that they have opened a new plan before the scientific community. In addition, they have shown the IT community that optimization problems can be solved effectively by hybridizing and combining existing algorithmic structures, rather than by using existing paradigms. An important contribution of these has been to offer a new perspective in algorithmic design. A solution can be generated by combining the forces of different paradigms and obtaining one that is able to overcome each paradigm separately. It is the basis of a problem-oriented algorithmic design, which is the natural consequence of free lunch theorems. The founding concept of the automatic and real-time design of solutions to problems will probably be the future of computer information, because machines, in the future, will have to analyze and "understand" problems automatically, by proposing an optimal solution.

2. OBJECTIVES

The aim of this paper is to present the hybridization method, and hybrid algorithms, especially memetic algorithm, so that investors must find quick solutions to make the right decision in a short time, so as to build forecast models, make certain assumptions for implicit variables, which are difficult to understand. To improve the prediction process, researchers have developed hybrid models for forecasting. The purpose of this paper is to present hybrid methods. Moreover, another goal is to use other hybrids to have the desired results.

3. METHODOLOGY

Memetic algorithms are population-based metaheuristics. The earliest implementation of memeticia algorithms was given in the context of Traveling Salesman Problem (TSP). The concept of meme is borrowed from philosophy and is intended as a unit of cultural transmission. So complex ideas can be broken down into memes that spread and move within a population. In this way, the culture evolves and tends towards progressive improvements. Strong ideas resist over time and are propagated within a community, and weak ideas are no longer remembered and tend to disappear.

The notion of "memetic algorithms" is used to include a wide class of metaheuristics, which can be considered general purpose methods. That said, the method uses a multitude of agents and has proven to be lucky in a wide range of difficult areas, especially for optimization issues. Unlike traditional methods of evolutionary computation, memetic algorithms are concerned with exploiting all available knowledge about the problem studied. Incorporating knowledge of the problem area is not an optional mechanism, but is a fundamental feature that characterizes memetic algorithms. This functional philosophy illustrates in a complex way the term "memetic". Founded by Dawkins, the word "meme" denotes an analogy with the gene in the context of cultural evolution.

Dawkins said: "Examples of memories are songs, ideas, catch phrases, clothing fashion, ways to make vessels or build arches. Just as genes spread in the gene pool by jumping from body to body through sperm or eggs, likewise memes propagate in the meme pool, jumping from brain to brain through a process that, in a broad sense, can be called imitation. " This characterization of a meme denotes that, in the processes of cultural evolution, information is transmitted unaltered between individuals. But, it is processed and improved by the communicating parties. This improvement is achieved in memetic algorithms by incorporating heuristics, approximation algorithms, local search techniques, specialized recombination operators, exact methods and much more. Thus, most memetic algorithms can be interpreted as a search strategy in which a population of optimization agents cooperates and competes. The fame of memetic algorithms can be explained as a direct consequence of the synergy of the different searches it includes.

The most complex feature of memetic algorithms, the inclusion of knowledge of the problems mentioned above, is proved by strong theoretical results. Initially, Hart and Belew deduced, then Wolpert and Macready later popularized the so-called free lunch theorem, which is a search algorithm that meets strictly according to the quantity and quality of knowledge of the problem they incorporate. Due to the fact that the term hybridization is used to represent the process of incorporating knowledge of problems, it follows that evolutionary algorithms are also called "hybrid evolutionary algorithms". One of the first memetic algorithms, one dates from 1988 and has been considered by many researchers a hybrid of traditional genetic algorithms and annealing simulation. The reason was to find a way out of the limits of both techniques to a well-studied combinatorial optimization problem, the min euclidean traveling salesman (MIN ETSP) problem. According to the authors, the debut inspiration came from the computer game tournaments used to study the "evolution of cooperation". This approach had several features that anticipated many current algorithms. The competition phase of the algorithm was based on the new allocation of search points in the configuration phase, a process that includes a "fight" for survival, followed by socal cloning, which has a strong resemblance to the "go with the winners" algorithms. The cooperation phase followed by the local search can be called "go with the local winners", because the optimization agents were arranged with a topology of a two-dimensional toroidal lattice.

Later, Moscato and Norman found that they shared similar views with other researchers proposing "island models" for memetic algorithms. Spatialization is now recognized as the "catalyst" responsible for a variety of phenomena. Then, in 1989, they identified several authors who also pioneered the introduction of heuristics to improve solutions before recombining them. Coming mainly from the field of GA, several authors have introduced knowledge of the field with problems in a variety of ways. In On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms (1989) by Moscato, the name "memetic algorithms" was first introduced. It has also been pointed out that cultural evolution can be a better working metaphor for these metaheuristics to avoid "biologically restricted" thinking that restricts progress at that time. So, resuming, a memetic algorithm is a hybridization between a genetic algorithm and a local search method. More over, the power of genetic algorithms comes from the fact that they are capable of searching globally the solutions space. The mutation operator modifies the individual. The operator diversifies the individuals while the selection is responsible for retaining the best individuals.So, genetic algorithms are beeing repeatedly hybridized with local research methods. These methods, genetic algorithm and a local search, are complementary because one makes it possible to detect important regions in the search space while the other focuses intensively on exploring these areas of the search space of solutions.

Moscato's idea is to supplement a local search procedure which can be a descent method or a more far gonefar gone local search. This procedure will be applied to any new individual obtained during the research. So, by changing leads to profound modifications in the algorithm. It results a new individual from two selected parents and under certain conditions it is applied a mutation operator to this individual.

In the algorithm, the mutation operator provides the method diversification and the intensification is produced by the application of the local search method.The different steps of the memetic algorithm are :

Initialize: generate an initial population $\boldsymbol{P}$ of solutions
Apply a local search procedure on each $\boldsymbol{P}$ solution
Repeat
Selection: choose two solutions x and x'
Crossing : combining two parent solutions x and x' to form a solution y
Local search: apply a local search procedure on $\boldsymbol{y}$
Mutation: apply a mutation operator on
Choose an individual y' to be replaced in the population
Replace y' by y in the population
Until satisfying a stop criterion.

4. ANALYSES

The performance of a memetic algorithm depends to a large extent on the correct choice of local search strategies (memes), on the identification of the subset under local improvements and on the convergence criterion used in local search strategies. Below is a study designed to solve numerical optimization problems constrained by traditional representation. In this case, a local search is embedded in an evolutionary algorithm to accelerate its convergence rate.

Different complex design and decision processes require a solution to restricted optimization problems (ConOP). Furthermore, ConOPs can be defined mathematically as follows:

$$\begin{aligned}&\text{Minimize } f(\mathrm{X})\\&\text{Subject to } g_i(\mathrm{X}) \geq 0, i = 1,\ldots,m\\&h_j(\mathrm{X}) = 0, i = 1,\ldots,p\\&L_i \leq x_i \leq U_i, i = 1,2,\ldots n\end{aligned}$$

where X = (x_1,..., x_n) is a vector with n decision variables, f (X) is the objective function, g_i(X) is the i^{th} inequality constraint, h_j(X) is the j^{th} equality constraint, each x_i has a lower limit L_i and an upper limit U_i.

Based on mathematical properties, ConOPs can be of several types. They contain various variables, such as real, integer and discrete, and may have equality and / or inequality constraints. Objective and constraint functions can be linear / nonlinear. The problem can have one or more objectives (which can be maximizing or minimizing). Functions can be continuous / discontinuous and unimodal / multimodal. The application of restricted optimization methods is wide. Examples include: planning (resource allocation, logistics, production planning and scheduling), engineering design (welded beam, pressure vessel), medical science (beam optimization for radiotherapy, DNA sequencing) and computer science (database design and data extraction) . Researchers and practitioners use both conventional methods of mathematical optimization and newer methods, relying on computational intelligence to solve ConOP. A disadvantage of conventional optimization methods is that they require specific properties (such as convexity, continuity, and differentiality) of the mathematical model and thus require simplifications of the problem by assumptions. Moreover, the choice of a method is determined by a multi-classification of problems.

The memetic algorithms used for constrained optimization can be highlighted by the classification presented in the figure below:

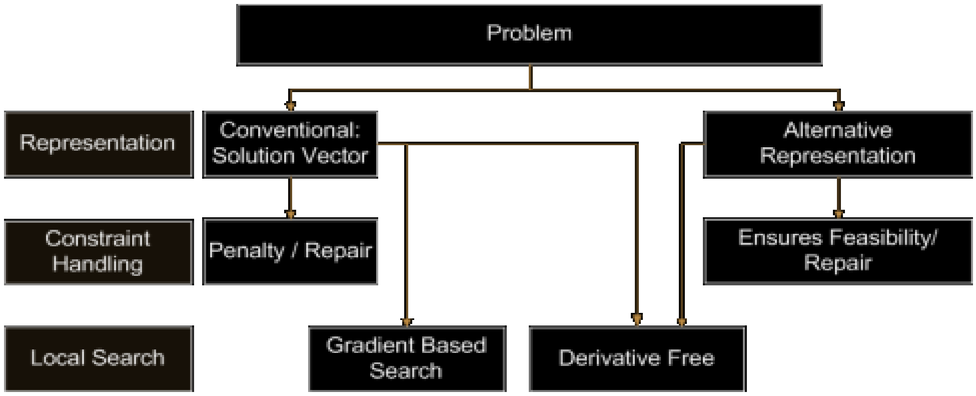

Figure 1: Classification of Memetic Algorithms[10]

To illustrate a numerical case, the Infeasibility of the Empowered Memetic Algorithm for Constrained Optimization Problems is presented: MA with conventional representation. So, this is the Infeasibility Empowered Memetic Algorithm (IEMA), which is a combination of the Infeasibility Driven Evolutionary Algorithm (IDEA) and a local search based on the Sequential Quadratic Program (SQP). IDEA is a derivative variant of evolutionary algorithms in which a small proportion of marginally unviable solutions are kept to accelerate the convergence rate. More over, the optimal unviable solutions are ranked higher than the feasible solutions and therefore the search is carried out in both feasible and ineffable regions; resulting in a higher convergence rate to optimal solutions.

Inflatability-driven evolutionary algorithm (IDEA)

A single-objective optimization problem can be formulated as shown in the equations:

$$\text{Minimize } f(\mathrm{X})$$
$$\text{Subject to } g_i(\mathrm{X}) \geq 0, i = 1, \ldots, m$$

$$h_j(\mathrm{X}) = 0, i = 1, \ldots, p$$
$$L_i \leq x_i \leq U_i, i = 1,2, \ldots n$$

To efficiently search the design space, the optimization problem is reformulated as a bi-objective optimization problem:

$$Minimize\ f_1'(x) = f_1(x)$$
$$f_2'(x) = violation\ measure$$

The target is a measure of breach of coercion, called a 'breach measure'. Each solution of the population is assigned m ranks, corresponding to each m constraint. To obtain the ranks corresponding to constraint i, all solutions are sorted according to the value of the constraint violation. Solutions that do not violate the constraint are assigned a rank of 0. The solution with the lowest value of the constraint violation is given a rank of 1, and the rest of the solutions are assigned ascending ranks in ascending order of their values of the violation of the constraint. The process is repeated for all constraints and each solution in the population receives m ranks. The measure of the violation is the sum of these m ranks corresponding to the constraints m.

The main steps of IDEA are presented in the algorithm below.

Infeasibility Driven Evolutionary Algorithm (IDEA) [10]

begin

// Given population size *N* number of generations $N_G > 1$ and Proportion of infeasible solutions $0 < \alpha < 1$

$N_{inf} \leftarrow \alpha * N;$

$N_f \leftarrow N - N_{inf};$

$set\ pop_1 \leftarrow Initialize();$

$Evaluate(pop_1);$

$for\ i = 2\ to\ N_G\ do$

$child\ pop_{i-1} \leftarrow Evolve(pop_{i-1});$

$Evaluate(child\ pop_{i-1});$

$(S_f, S_{inf}) \leftarrow Split(pop_{i-1} + child\ pop_{i-1});$

$Rank(S_f);$

$Rank(S_{inf});$

$pop_1 \leftarrow S_{inf}(1 : N_{inf}) + S_f(1 : N_f)$

$endfor$

end

The next generation solutions are selected from both sets to maintain unviable solutions in the population. Moreover, unviable solutions are ranked higher than feasible solutions, to ensure a selection pressure in the need to create better solutions, which leads to an active search through the unviable search space.

Infeasibility Empowered Memetic Algorithm (IEMA)

This IEMA algorithm is realized using IDEA as the basic algorithm.

For unique objective issues, a local search can be a very effective tool for optimization. However, its performance depends largely on the boot solution. The proposed algorithm tries to exploit the advantages of both approaches. IEMA is presented in the algorithm below.

Infeasibility Empowered Memetic Algorithm (IEMA)[10]

begin

// Given population size N number of generations $N_G > 1$ and Proportion of infeasible solutions $0 < \alpha < 1$

$N_{inf} \leftarrow \alpha * N;$

$N_f \leftarrow N - N_{inf};$

$pop_1 = Initialize();$

$Evaluate(pop_1);$

$for\ i = 2\ to\ N_G\ do$

$child\ pop_{i-1} \leftarrow Evolve(pop_{i-1});$

$Evaluate(child\ pop_{i-1});$

$(S_f, S_{inf}) \leftarrow Split(pop_{i-1} + child\ pop_{i-1});$

$Rank(S_f);$

$Rank(S_{inf});$

$pop_1 \leftarrow S_{inf}(1 : N_{inf}) + S_f(1 : N_f)$

x ← Random solution in pop_i;

x_{best} ← Local_search (**x**);

// x_{best} is the best solution found using local search from x

Replace worst solution in pop_i with x_{best};

Rank(pop_i);

Rank the solutions again in pop_i

$endfor$

end

In IEMA, in each generation, in addition to the evolution of solutions in IDEA, a local search is made from a random solution from the population, for a prescribed number of evaluations of functions.

Results on CEC-2010 Benchmark Problems

The performance of IEMA is presented for one of the most recent difficult sets of narrow optimization references, namely that of the IEEE CEC-2010, restricted optimization competition. The parameters used for IEMA are the same for each problem, ie the parameters are not adjusted throughout the problems. The parameters are shown in the figure below.

Parameter	Value
Population Size	200
Max. FES	for 10D problems: 200000
	for 30D problems: 600000
Crossover Probability	0.9
Crossover index	15
Mutation Probability	0.1
Mutation index	20
Infeasibility Ratio (α)	0.9

Figure 2 Parameters used for IEMA[10]

The results of problems 10D are shown in Figure 3, while the results of problems 30D are listed in Figure 4. To determine the median, the following procedure is adopted. All runs in which a feasible solution has been found are sorted according to the best value of the function obtained. After that, all runs in which no feasible solution is found are sorted according to the average violation of the constraint of the best solution obtained. Feasible runs are ranked above impossible turnovers. In the sorted list, the 13th solution is reported as the median solution (only if the median is feasible). The best, average and worst runs reported in the tables are based only on runs in which at least one possible solution has been found.

	C01	C02	C03	C04	C05	C06
Best	-0.74731	-2.27771	1.46667e-16	-9.98606e-06	-483.611	-578.662
Median	-0.74615	-2.27771	3.2005e-15	-9.95109e-06	-483.611	-578.662
Mean	-0.743189	-2.27771	6.23456e-07	-9.91135e-06	-379.156	-551.47
std	0.00433099	1.82278e-07	1.40239e-06	8.99217e-08	179.424	73.5817
Feasible	25	25	25	25	24	24

	C07	C08	C09	C10	C11	C12
Best	1.74726e-10	1.00753e-10	1.20218e-09	5.4012e-09	-0.00152271	-10.9735
Median	1.9587e-09	3.94831e-09	333.32	42130.4	-0.00152271	-0.199246
Mean	3.25685e-09	4.0702	1.95109e+12	2.5613e+12	-0.00152271	-0.648172
std	3.38717e-09	6.38287	5.40139e+12	3.96979e+12	2.73127e-08	2.19928
Feasible	25	25	23	19	24	24

	C13	C14	C15	C16	C17	C18
Best	-68.4294	8.03508e-10	9.35405e-10	4.44089e-16	9.47971e-15	2.23664e-15
Median	-68.4294	1.29625e-08	26.1715	0.0320248	2.59284e-12	6.78077e-15
Mean	-68.0182	56.3081	1.57531e+08	0.0330299	0.00315093	1.61789e-14
std	1.40069	182.866	6.04477e+08	0.0226013	0.0157547	3.82034e-14
Feasible	25	25	25	25	25	25

Figure 3: Performance of IEMA on 10D problems[10]

	C01	C02	C03	C04	C05	C06
Best	-0.821883	-2.28091	-	-	-286.678	-529.593
Median	-0.819145	-2.27767	-	-	-	-
Mean	-0.817769	-1.50449	-	-	-270.93	-132.876
std	0.00478853	2.14056	-	-	14.1169	561.042
Feasible	25	25	0	0	4	2

	C07	C08	C09	C10	C11	C12
Best	4.81578e-10	1.12009e-09	7314.23	27682	-	-
Median	6.32192e-10	0.101033	7.91089e+06	1.1134e+07	-	-
Mean	8.48609e-10	17.7033	2.98793e+07	1.58342e+07	-	-
std	4.84296e-10	40.8025	4.50013e+07	1.68363e+07	-	-
Feasible	25	25	25	25	0	0

	C13	C14	C15	C16	C17	C18
Best	-68.4294	3.28834e-09	31187.6	6.15674e-12	9.27664e-10	1.37537e-14
Median	-67.6537	7.38087e-09	7.28118e+07	1.26779e-10	5.67557e-06	2.12239e-14
Mean	-67.4872	0.0615242	2.29491e+08	0.00163294	0.0883974	4.73841e-14
std	0.983662	0.307356	4.64046e+08	0.0081647	0.15109	6.5735e-14
Feasible	25	25	25	25	25	25

Figure 4: Performance of IEMA on 30D problems[10]

Plots show feasible solutions only for the best turnovers that correspond to these problems. Objective values were represented in the journal scale to aid visualization.

The time complexity of the algorithm is shown in the figure below.

	$T1$	$T2$	$(T2-T1)/T1$
10D problems	2.57636	9.05104	2.51312
30D problem	2.57854	13.2825	4.1512

Figure 5: Time complexity of IEMA (in seconds) [10]

5. CONCLUSIONS

The methods of optimization founded on the fundamental metaheuristics characteristics are useful in difficult optimization without the need to modify the basic structure of the used algorithm. They have become very accesible due to their ease of use in various fields.It should be noted that good performance often requires a convenient formalization of the given problem and a superior adaptation of a metaheurist. Notwithstanding the remarkable success of their approach, metaheuristics present difficulties that the user faces in the case of a concrete problem, such as selecting an effective method to have an optimal solution and adjusting parameters that can be met in theory but impractical in practice. Researchers are willing to exceed these problems by improving techniques, including the metaheuristics hybridization. This hybridization capitalize the power of differents algorithms and by mixing them results a single meta-algorithm.

So, the major difficulty had by a researcher in the presence of a optimization problem is that of selecting an efficient method able to produce an optimal solution by having an agreable quality. Therefore, the theory is not yet of much help, because the convergence metaheuristics theorems are sometimes inexistent or implementable under very limitative hypotheses.

Furthermore, the optimal adjustment of the different metaheuristic parameters, is possible in theory, but is frequently not applicable in practice, because it etablishes a prohibitive cost of calculation.

Better than this, they depend very much on the optimization problem given and change from one problem to another.

The studied memetic algorithm takes over a conventional representation system and combines a global population-based search and an SQP for local search. The population-based global search constituent of the memetic algorithm naturally maintains a fraction of the marginally unviable solutions in an attempt to increase the convergence rate.

REFERENCES

1. Auad, M.I. Conception d'Algorithmes Hybrides pour l'Optimisation de l'Energie Memoire dans les Systemes Embarques et de Fonctions Multimodales, University Henri Poincare – Nancy, 2011
2. Bensaifia, A.N. Methode hybride basse sur les algorithms a evolution

differentielle, Bounaama University, Alger, 2018

3. El Dor, A. — Perfectionnement des algorithmes d'optimisation par essaim particulaire: applications en segmentation d'images et en électronique, University Paris-Est, 2012
4. Eusuff M., Lansey K., Pasha F. — Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Engineering Optimization Vol. 38, No. 2, Arizona, 2006.
5. Hachimi, H. — Hybridations d'algorithmes metaheuristiques en optimisation globale et leurs applications, INSA de Rouen; Ecole Mohammadia d'ingenieurs, Maroc, 2013
6. Houbad, Y., Hassam, A., Souier, M., Sari, Z. — Adaptation d'un algorithme mémétique pour l'amélioration des performances d'un système flexibles de production, University Abou-Bakr Belkaid B.P. 119 Tlemcen, Alger.
7. Hemmak, A. — Support de cours d'optimisation combinatoire, Alger, 2017;
8. Nazmul Siddique, Hojjat Adeli. — Computational intelligence : synergies of fuzzy logic, neural networks, and evolutionary computing, A John Wiley & Sons, Ltd., Publication, 2013;
9. Moscato P., Cotta C. — A modern introduction to memetic algorithms, Springer Science and Business, 2010;
10. Neri F. Moscato P, Cotta C. — Handbook of memetic algorithms, Springer, Warsaw, 2010;
11. Schumann E., Maringer D.,Gilli M. — Numerical Methods and Optimization in Finance, Switzerland,Academic Press Elsevier, 2011.