

On heuristics as a fundamental constituent of soft computing

José L. Verdegay^{a,*}, Ronald R. Yager^b, Piero P. Bonissone^c

^a*Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 18071 Granada, Spain*

^b*Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, USA*

^c*GE Global Research, Building K1, 4A10A, 1 Research Circle, Schenectady, NY 12309, USA*

Received 26 February 2007; received in revised form 8 August 2007; accepted 27 August 2007
Available online 6 September 2007

Abstract

Although as such one dates back the idea of setting the area of soft computing to 1990, it was in 1994 that L.A. Zadeh established his worldwide accepted definition of soft computing. As it is well known since the seminal definition of a fuzzy set, different equivalent definitions of the concept have been proposed, analyzed and used. But, in spite of the former main constituents could be currently others and hence they should be revised, and the same cannot be said of soft computing. From this point of view, in order to narrow this gap, in this paper the role played until now by these main soft computing ingredients is analyzed, and then an original proposal of the new constituents, mainly focused on the introduction of the broader topic of metaheuristics instead of evolutionary algorithms, is justified, presented and described.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Soft computing; Fuzzy sets; Probabilistic reasoning; Neural networks; Metaheuristics

1. Introduction

Ever since L.A. Zadeh introduced fuzzy sets in 1965 [23], applications and developments based on this simple concept have evolved to such an extent that it is practically impossible nowadays to encounter any area or problem where applications, developments, products, etc. are not based on the fuzzy sets.

One important type of problems in particular are the optimization problems, which optimize the value that a function may reach on a previously specified set, and these and everything relating to them are covered by the area known as *mathematical programming*. When fuzzy elements are considered in mathematical programming, fuzzy optimization methods emerge, and these are perhaps one of the most fruitful areas of fuzzy-related knowledge, both from the theoretical and the applied points of view. Yet despite all its methods and models for solving the enormous variety of real practical solutions, as with conventional mathematical programming, it cannot solve every possible situation for, while a problem may be expressed in fuzzy terms, it cannot be solved only with fuzzy techniques.

The ease of resolving ever larger real problems, the impossibility of discovering exact solutions to these problems in every case and the need to provide answers to the practical situations considered in a great many cases have led to the increasing use of heuristic-type algorithms which have proved to be valuable tools capable of providing solutions

* Corresponding author.

E-mail address: verdegay@decsai.ugr.es (J.L. Verdegay).

where exact algorithms are not able to. In recent years, a large catalogue of heuristic techniques has emerged inspired by the principle that satisfaction is better than optimization, or, in other words, rather than not being able to provide the optimal solution to a problem, it is better to give a solution which at least satisfies the user in some previously specified way, and these have proved to be extremely effective.

These heuristics are said to have been mostly inspired by nature, society, physics, etc. to produce theoretical models which match the circumstances considered, and, from this perspective, it has been possible to solve cases which, until only very recently, were impossible with conventional techniques. In most cases, however, the solutions achieved have not been optimal and are instead “almost optimal”, having been obtained with criteria other than the classic “achieving the best value of the objective function”, by considering characteristics which have been subjectively established by the decision-maker [2].

It is well known that when we speak of human subjectivity, or even closeness to an ideal value, the best comparative way of modelling this type of situation is by means of *fuzzy* sets, or more generally with soft computing methodologies. This method of modelling subjectivity (which is so developed in other fields) has hardly ever been applied to the case of heuristic algorithm design despite all indications that this might well be a very promising approach [18], because in addition to providing solutions which are as close to the optimum as other well-known conventional heuristic ones:

- (a) soft computing-based heuristics may solve the problem in a less costly way than other methods;
- (b) they generalize already known heuristics and
- (c) the hybridization in the soft computing context favors and enriches the appearance of original procedures which can help resolve new problems.

However, while the historic path of fuzzy sets and systems has been much explored, the same cannot be said of soft computing. In order to narrow this gap, we will describe what soft computing is and what is understood by heuristics, and from both concepts we will attempt to find a common ground where the best of both the worlds can be combined. There will be results: the first is that there will be soft computing-based metaheuristic (MH) procedures which appear to be among the most promising tools for the effective solution of problems which are as yet impossible to solve, and also for finding solutions which suit the person looking for them; and the second (as a result of the first) is that a new description will emerge of the components which define soft computing and this will further extend the application sphere.

Consequently, the next section presents the former concept of soft computing and its main classical constituents. Then Section 3 focuses on the definition of the heuristics and the MHs. The review of the soft computing components is carried out in Section 4, and in Section 5 new hybrid MHs in soft computing are presented and briefly described. The main conclusions and bibliography close the paper.

2. Soft computing

Prior to 1994 when Zadeh [24] first defined “soft computing“, the currently handled concepts used to be referred to in an isolated way, whereby each was spoken of individually with an indication of the use of fuzzy methodologies. Although the idea of establishing the area of soft computing dates back to 1990 [26], it was in [24] that Zadeh established the definition of soft computing in the following terms:

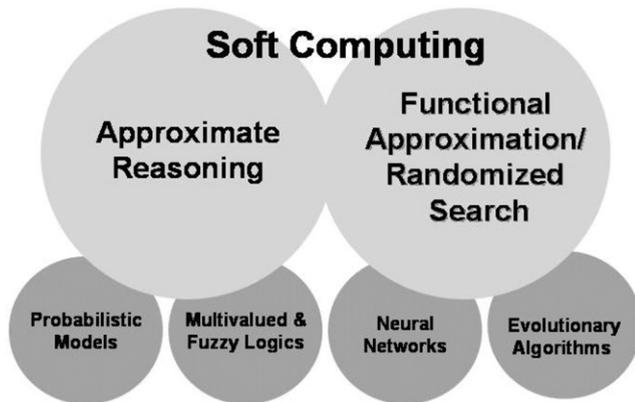
Basically, soft computing is not a homogeneous body of concepts and techniques. Rather, it is a partnership of distinct methods that in one way or another conform to its guiding principle. At this juncture, the dominant aim of soft computing is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solutions cost. The principal constituents of soft computing are fuzzy logic, neurocomputing, and probabilistic reasoning, with the latter subsuming genetic algorithms, belief networks, chaotic systems, and parts of learning theory. In the partnership of fuzzy logic, neurocomputing, and probabilistic reasoning, fuzzy logic is mainly concerned with imprecision and approximate reasoning; neurocomputing with learning and curve-fitting; and probabilistic reasoning with uncertainty and belief propagation.

It is therefore clear that rather than a precise definition for soft computing, it is instead defined by extension, by means of different concepts and techniques which attempt to overcome the difficulties which arise in real problems which occur in a world which is imprecise, uncertain and difficult to categorize.

There have been various subsequent attempts to further fit this definition, with differing results, and among the possible alternative definitions, perhaps the most suitable is the one presented in [13]: “Every computing process that purposely includes imprecision into the calculation on one or more levels and allows this imprecision either to change (decrease) the granularity of the problem, or to “soften” the goal of optimization at some stage, is defined as to belonging to the field of soft computing”.

The viewpoint that we will consider here (and which we will adopt in the future) is another way of defining soft computing, whereby it is considered to be the antithesis of what we might call *hard computing*. This viewpoint is consistent with the one in [25,26]. Soft computing could therefore be seen as a series of techniques and methods so that real practical situations could be dealt with in the same way as humans deal with them, i.e. on the basis of intelligence, common sense, consideration of analogies, approaches, etc. In this sense, soft computing is a family of problem-resolution methods headed by approximate reasoning and functional and optimization approximation methods, including search methods. Soft computing is therefore in the theoretical basis for the area of intelligent systems.

From this other viewpoint on a second level, soft computing can be then expanded into other components which contribute to a definition by extension, such as the one first given. From the beginning [3], the components considered to be the most important in this second level are probabilistic reasoning, fuzzy logic and fuzzy sets, neural networks and genetic algorithms (GA), which because of their interdisciplinary, applications and results immediately stood out over other methodologies such as the previously mentioned chaos theory, evidence theory, etc. The popularity of GA, together with their demonstrated efficiency in a wide variety of areas and applications, their attempt to imitate natural creatures (e.g. plants, animals, humans) which are clearly soft (i.e. flexible, adaptable, creative, intelligent, etc.), and especially the extensions and different versions, transforms this fourth second-level ingredient into the well-known evolutionary algorithms (EA) which consequently comprise the fourth fundamental component of soft computing, as shown in the following diagram:



From this last conception of soft computing, fuzzy sets and fuzzy logic playing a necessarily basic role, we can describe other areas emerging around it simply by considering some of the possible combinations which can arise [3,5]:

1. From the first level and beginning with approximate reasoning methods, when we only concentrate on probabilistic models, we encounter the Dempster–Shafer theory and Bayesian networks. However, when we consider probabilistic methods combined with fuzzy logic, and even with some other multi-valued logics, we encounter what we could call hybrid probabilistic models, fundamental probability theory models for fuzzy events, fuzzy event belief models and fuzzy influence diagrams.
2. When we look at the developments directly associated with fuzzy logic, fuzzy systems and in particular fuzzy controllers stand out. Then, arising from the combination of fuzzy logic with neural networks and EA are the fuzzy logic-based hybrid systems, the foremost exponents of which are the fuzzy neural systems, the controllers adjusted by neural networks (neural fuzzy systems which differ from the previously mentioned fuzzy neural systems) and the fuzzy logic-based controllers which are created and adjusted with EA.
3. Moving through the first level to the other large area covered by soft computing (functional approach/optimization methods) the first component which appears is that of neural networks and their different models. Arising from the interaction with the fuzzy logic methodologies and the EA methodologies are the hybrid neural systems, and

in particular the fuzzy control of network parameters, and the formal generation and the weight generation in neural networks.

4. The fourth typical component of soft computing and perhaps the newest yet possibly most up-to-date is that of EA, and associated with these are four large, important areas: evolutionary strategies, evolutionary programming, GA and genetic programming. If we were only to focus on these last areas, we could consider that in this case the amalgam of the methodologies and the techniques associated with soft computing culminate in three important lines: fuzzy genetic systems, bioinspired systems and applications for the fuzzy control of evolutionary parameters.

On further examination of this last component some additional considerations are needed. Firstly, independent of the broad-minded approach adopted to contemplate what can be embraced by the fuzzy genetic systems, the bioinspired systems and the fuzzy control applications on evolutionary parameters, other important topics are missing from this description. Secondly, if we are referring in particular to the bioinspired systems, it is clear that not only are they the product of fuzzy logic, neural networks or EA (with all the variants that we can consider for these three components) but also that other extremely important methodologies are involved in them.

In the sections which follow we will therefore justify a new definition for soft computing components, which were first referred to in [19], in order to provide a clearer perspective of the different areas that these cover without any loss of essence.

3. Heuristics and MHs

As stated in [18], since the *fuzzy boom* of the 1990s, methodologies based on the fuzzy sets (i.e. soft computing) have become a permanent part of all areas of research, development and innovation, and their application has been extended to all areas of our daily life: health, banking, home, and are also the object of study on different educational levels. Similarly, there is no doubt that, thanks to the technological potential that we currently have, computers can handle problems of tremendous complexity (both in comprehension and dimension) in a wide variety of new fields.

As we mentioned above, since the mid-1990s, GA (or EA from a general point of view) have proved to be extremely valuable for finding good solutions to specific problems in these fields, and, thanks to their scientific attractiveness, the diversity of their applications and the considerable efficiency of their solutions in intelligent systems, they have been incorporated into the second level of soft computing components.

EA, however, are merely another class of heuristics, or MHs, in the same way as taboo search, simulated annealing, hill climbing, variable neighborhood search, estimation distribution algorithms (EDA), scatter search, GRASP, reactive search and very many others are. Generally speaking, all these heuristic algorithms (MHs) usually provide solutions which are not ideal, but which largely satisfy the decision-maker or the user. When these act on the basis that satisfaction is better than optimization, they perfectly illustrate Zadeh's famous sentence [24]: "...in contrast to traditional hard computing, soft computing exploits the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution-cost, and better rapport with reality".

Consequently, among the soft computing components, instead of EA (which can represent only one part of the search and optimization methods used), heuristic algorithms and even MHs should be considered.

There is usually controversy about the difference between MHs and heuristics, and, while it is not our intention here to enter into this debate, we are interested in offering a brief reflection on both concepts. The term *heuristics* comes from the Greek word "heuriskein", the meaning of which is related to the concept of finding something and is linked to Archimedes' famous and supposed exclamation, "Eureka!".

On this basis, a large number of heuristic procedures have been developed to solve specific optimization problems with a great success, and the best of these have been extracted and used in other problems or in more extensive contexts. This has contributed to the scientific development of this field of research and to the extension of the application of its results. As a result, MHs have emerged, a term which appeared for the first time in an article by Glover in 1986 [9].

The term MHs is derived from the combination of the word *heuristics* with the prefix *meta* (meaning beyond or of a higher level), and, although there is no formal definition for the term MHs, the following two proposals give a clear representation of the general notion of the term:

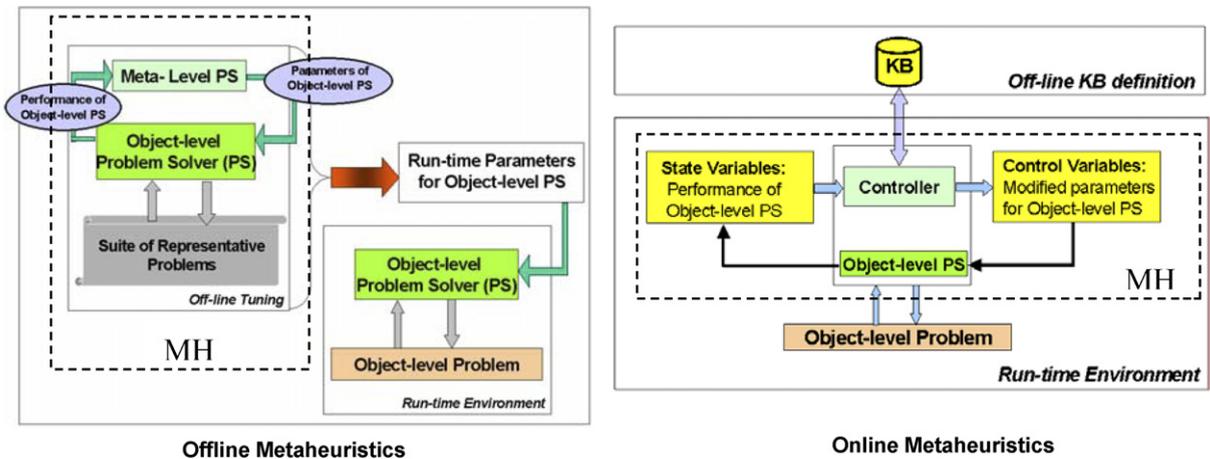
- (a) Osman and Laporte [15]: "An iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space".

- (b) Voss et al. [22]: “An iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high quality solutions”.

It is clear that MHs are more broad-brushed than heuristics. The limitation in the above definitions consists in assuming that MHs are a master process to generate, search, guide or control *subordinate heuristics*. In the broader framework of SC, the subordinate processes do not need to be heuristics, but are in fact SC components (which might include global search methods, such as EA). We prefer to use the etymological sense of *meta*, as “*of a higher level*”, a connotation that is widely used in logic, reasoning and AI, therefore, we propose a formal distinction between *meta-level* and *object level*. At the object level we have problem-solver models based on any technique but, within the scope of this paper, based on SC components. At the meta-level we have the designers, controllers or guides of the object-level models.

Within this definition of MHs, we would also like to distinguish between *offline* and *online* MHs. Offline MHs deal with the design of the object-level model—once the design is complete, a run-time object-level model is generated and used to solve the problem without any further modification. Online MHs deal with the monitoring, guidance and control of the run-time model.

The following two figures illustrate the use of offline and online MH for parameter design and control. Similar MHs could be built for structure design and modification of the object-level models. In summary, we could define MHs as the process to generate and search in the object-level model design space (for offline MH), or to assess and modify in the object-level model behavior/control space (for online MH) to improve the overall object-level model performance. For a more detailed explanation, please refer to Refs. [4,6].



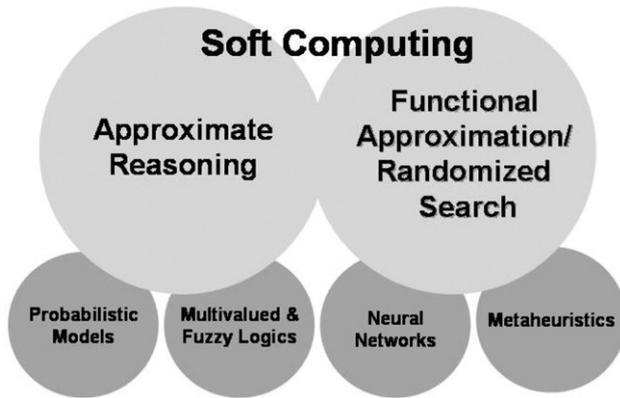
In the sections which follow, we will focus on the concept of MHs, and will start by pointing out that, in the terms that we have defined, certain MHs will always be better than the others in terms of their performance when it comes to solving problems.

In order to achieve the best performance of the MHs, it is desirable for them to have a series of “good properties” which include simplicity, independence, coherence, effectiveness, efficiency, adaptability, robustness, interactivity, diversity and autonomy [14]. In view of their definition and the series of desirable characteristics, it is both logical and obvious that EA are to be found among MHs and they are therefore well placed with the other second-level soft computing components to facilitate the appearance of new theoretical and practical methodologies, outlines and frameworks for a better understanding and handling of generalized imprecision in the real world (as explained in [26]).

4. A review of soft computing components

Returning to the previous description of the components which describe soft computing on the different levels, we could say that the most important second-level components are probabilistic reasoning, fuzzy logic and sets, neural networks and, in view of what we have explained, MHs (which would typically encompass EA but would not be confined to these exclusively). The new defining framework for the main methodologies which make up soft computing would

therefore be described as in the following diagram:

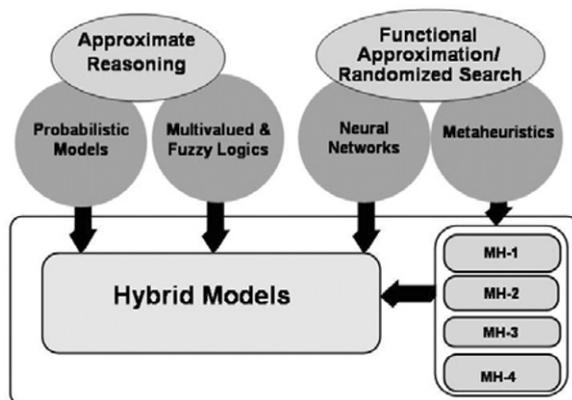


As we explained before, rather than understanding soft computing methodologies in an isolated way, it is necessary to understand them through the hybridization of their second-level components. Correspondingly, it is perfectly logical for us to explore the new theoretical–practical facets derived from the appearance of MHs among these components.

There are so many and such a variety of MHs available that it is practically impossible to agree on one universally accepted way of classifying them. Nevertheless, the hierarchy on which there is the most consensus considers three (or four) foremost groups:

- (1) MHs for evolutionary procedures based on sets of solutions which evolve according to natural evolution principles,
- (2) MHs for relaxation methods, problem-solving methods using adaptations of the original model which are easier to resolve,
- (3) MHs for neighborhood searches, which explore the solution space and exploit neighborhood structures associated with these solutions,
- (4) other types of intermediate MHs between the ones mentioned above or derived in some way from them, but which we will not consider because of their great variability (and to avoid dispersion).

We have decided to classify the MHs in this way, and what is at first apparent is that our previous definition of soft computing “by extension” according to its components not only maintains the essence of Zadeh’s original definition but generalizes and expands it to contemplate new possibilities. In effect, if we were to call these four groups of MHs MH(1), . . . , MH(4), respectively, the previous diagram could now be represented more explicitly as shown below:



where, due to the fact that there are still classic SC components, the different known and studied areas remain as they are, emerging as always when two or more of these components are interrelated with each other. However, as a result of having incorporated new possibilities into the fourth component (MHs), it now makes perfect sense to wait for new hybrid models to appear to be developed.

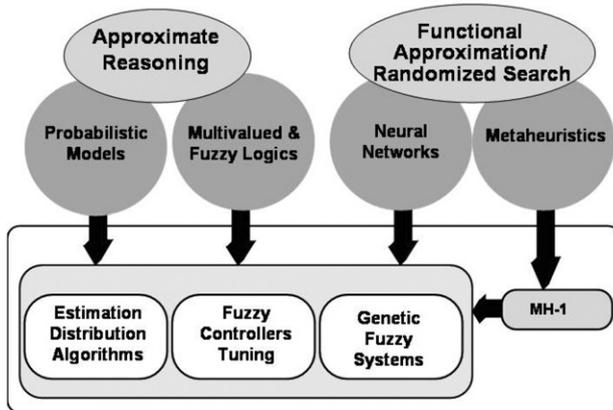
In order to demonstrate the range of study areas at our disposal when MHs are taken as the base component, in the following sections we will concentrate on describing the hybridizations which arise through the use of the previous categorization.

5. Hybrid MHs in soft computing

In this section, we will consider the three main previously mentioned groups of MHs. From these, we will then describe the new MHs which have emerged, briefly dwelling on the less developed or less popular ones because they are more recent.

5.1. Evolutionary MHs

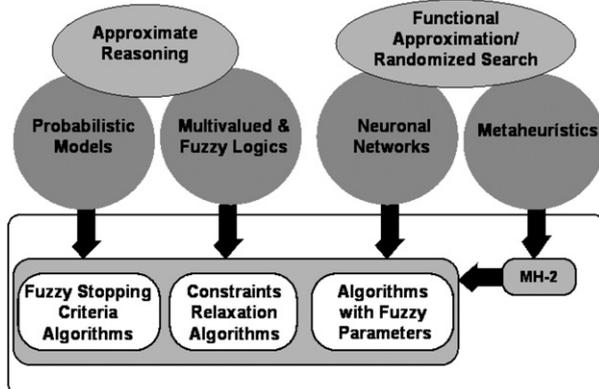
These MHs are by far the most popular and define mechanisms for developing an evolution in the search space of the sets of solutions in order to come close to the ideal solution with elements which will survive in successive generations of populations. In the context of soft computing, the hybridizations which take these MHs as a reference are fundamental:



Although this is a very important and a very wide area (covering everything from the fuzzy genetic systems to the adjustment of the fuzzy controllers with EA, to the evolution of Bayesian belief networks, to EDA, bioinspired systems, etc.), it is beyond the scope of this article and those interested should refer to [1,7,11,12]. Another historical reference is the concept of meta-GA, in which MH1 was used both at the meta-level and at the object level [10].

5.2. Relaxation MHs

A real problem may be relaxed when it is simplified by eliminating, weakening or modifying one of its characteristic elements. Relaxation MHs are strategies for relaxing the problem in heuristic design, and which are able to find solutions for problems which would otherwise have been very difficult to solve without the use of this methodology. Examples of these are rounding up or down or adjustments in nature, as occurring when an imprecisely and linguistically expressed quantity is associated with an exact numerical value. From this point of view, a real alternative is to flexibilize exact algorithms, introducing fuzzy stopping criteria [21,20], which eventually leads to rule-based relaxation MHs, admitting the vagueness of coefficients, justifying algorithms for resolving problems with fuzzy parameters and relaxing the verification of restrictions, allowing certain violations in their fulfillment.



In order to illustrate some of these MHs more specifically, we will consider algorithms with fuzzy stop criteria [21,20]. We know that the stop criteria fix the end conditions of an algorithm’s iterative procedure, establishing these criteria from the problem’s theoretical features, from the type of solution being sought and from the type of algorithm used. If a given algorithm provides the succession (x_n) of feasible solutions, some of the most frequent stop criteria are:

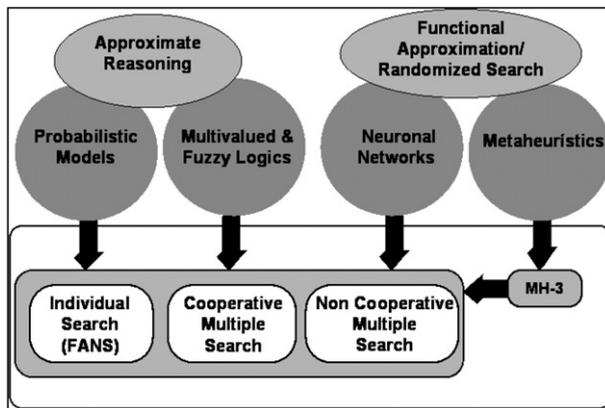
- (a) stop the process after N iterations;
- (b) stop the process when the relative or absolute distance between two elements in the succession from a certain iteration is less than or equal to a prefixed value;
- (c) stop the process when a prefixed measure $g(x_n)$ satisfies a certain condition such as being less than or equal to a constant.

In short, it can be said that an algorithm determines a reference set and stops when the set specified in the stop criteria has been obtained. The flexibilization of exact algorithms with the introduction of fuzzy stop criteria therefore assumes that the reference set is considered to be a fuzzy set, and the stop criteria are fixed according to the membership degree of the elements.

5.3. Search MHs

Generally speaking, these are probably the most important MHs, and their basic operation consists in establishing strategies for exploring the solution space of the problem and iterating the starting-point solutions. Although at first sight they might appear to be similar to the evolutionary search methods, they are not since the evolutionary search methods base their operation on the evolution of a population of individuals in the search space. These MHs are usually described by means of various metaphors, which classify them as bioinspired, sociological, based on nature, etc. and this makes them extremely popular.

However, outside this descriptive framework, given that a search can be made by means of a single search procedure (or by more than one in which case the search methods could either cooperate with each other or not) the search MH (without this classification being exclusive for this section) can be considered as individual or multiple, allowing in this last case the possibility for different agents to cooperate with each other. The different options which can emerge in the context of soft computing are collected in the following diagram:



Among the best known individual MHs are hill climbing, greedy like, multi-start, variable neighborhood, simulated annealing, taboo, etc. which have their own fuzzy extensions.

Independent of their specific method of action, all these MHs explore the search space according to evaluations of the objective function of the specific problem which is being solved, and this explicitly supposes performing numerical valuations with the help of an objective function in a precisely defined space. Only too often, however, the objective function represents some vaguely established property, and the search space (or the neighborhoods being searched) has no clearly defined boundaries, and this makes it logical to focus the application of these MHs with theoretical elements from the sphere of fuzzy logic and fuzzy sets. It is precisely in this context that FANS-type algorithms emerge [2,16].

FANS is a neighborhood search method where the solutions are evaluated not only in terms of the objective functions but also through the use of fuzzy properties and concepts which enable qualitative valuations on the solutions. It is also

a method which may be adapted to the context since its behavior varies according to the state of the search through the use of various administrators. FANS is based on four main components; operator (O), fuzzy valuation (FV), operator scheduler (OS) and neighborhood scheduler (NS), and a diagram of the algorithm is shown below to display the interaction between these four components:

```

Procedure FANS:
Begin
  While ( not-finalization ) Do
    /* the neighborhood scheduler is called */
    NS->Run(O,  $\mu()$ ,  $S_{cur}$ ,  $S_{new}$ ,  $ok$ );
    If ( $S_{new}$  is good enough in terms of  $\mu()$ ) Then
       $S_{cur} = S_{new}$ ;
      adaptFuzzyVal( $\mu()$ ,  $S_{cur}$ );
    Else
      /* No good enough solution was found */
      /* with the current operator */
      /* We change it with the operator scheduler */
      OS->Run(O);
    endif
    If (trappedCondition()) Then
      doRestart();
    endif
  endDo
End.

```

If, however, the search procedure is performed using various MHs, there is always the possibility of cooperation between these [8], and therefore the generalization of everything described so far to the context of parallelism, something which is obviously beyond the sphere of this paper but which is interesting to reflect on since, with the proliferation of parallel computing, more powerful work stations and faster communication networks, parallel implementations of MHs have emerged as something natural and provide an interesting alternative for increasing the speed of the search for solutions. Various strategies have correspondingly been proposed and applied and these have proved to be very efficient for resolving large-scale problems and for finding better solutions than those of their sequential counterparts due to the division of the search space, or because they have improved the intensification and diversification of the search. As a result, parallelism (and therefore multiple MHs) not only constitutes a way of reducing the execution times of individual MHs, but also of improving their effectiveness and robustness.

In the soft computing framework, the basic idea which has been developed so far has consisted in supposing that there is a set of resolving agents [17] which are basically algorithms for solving combinatorial optimization problems, and to execute them cooperatively by means of a coordinating agent to solve the problem in question, taking the generality based on the minimum knowledge of a problem as a fundamental premise. Each solving agent acts autonomously and only communicates with a coordinating agent to send it the solutions as it finds them and to receive guidelines about how to proceed. The coordinating agent receives the solutions found by each solving agent for the problem, and following a fuzzy rule base to model its behavior, it creates the guidelines which it then sends to them, thereby taking total control of the strategy.

6. Conclusion

The concept of a fuzzy set has been and is a paradigm in the scientific-technological world with important repercussions in all social sectors because of the diversity of its applications, of the ease of its technological transfer and of the economic saving that its use supposes. Although when the first article on the subject was published about 40 years ago it met with resistance from certain academic sectors, time has shown that fuzzy sets constitute the nucleus of a doctrinal body of indubitable solidness, dynamism and international recognition which is known as soft computing.

It is precisely this dynamism which has led us to reflect in this article on what the defining limits of soft computing are in an attempt to widen the range of its basic components with the inclusion of MHs. This wider and a more general

perspective of soft computing allows the possibility of incorporating new and as yet undeveloped search/optimization methods (without any of the already explored methods being the protagonist), thereby avoiding the tendency indicated by Zadeh in [26] to proclaim the methodology in which we are interested to be the best (which, as Zadeh pointed out, is yet another version of the famous hammer principle which says that “when the only tool you have is a hammer, everything begins to look like a nail”).

Acknowledgments

The authors are especially indebted to Didier Dubois both for his helpful criticism and comments and for a large number of excellent suggestions on a previous version of this paper, that undoubtedly helped to enhance the understanding of the original article. We gratefully acknowledge his support. The first author also acknowledged the support provided by the projects TIN2005-08404-CO4-01 (HeuriFuzzy) and TIC-00129 (MINAS).

References

- [1] M.G. Arenas, F. Herrera, M. Lozano, J.J. Merelo, G. Romero, A.M. Sánchez (Eds.), in: Proc. IV Spanish Congr. on Metaheuristics, Evolutionary Algorithms and Bio inspired (MAEB'05) I and II, 2005 (in Spanish).
- [2] A. Blanco, D. Pelta, J.L. Verdegay, A fuzzy valuation-based local search framework for combinatorial problems, *Fuzzy Optim. and Decision Making* 1 (2002) 177–193.
- [3] P. Bonissone, Soft computing: the convergence of emerging reasoning technologies, *J. Res. Soft Comput.* 1 (1) (1997) 6–18.
- [4] P. Bonissone, Soft computing and meta-heuristics: using knowledge and reasoning to control search and vice-versa, in: Proc. of the SPIE, Vol. 5200, Applications and Science of Neural Networks, Fuzzy Systems and Evolutionary Computation V, San Diego, CA, August 2003, pp. 133–149.
- [5] P. Bonissone, Y.-T. Chen, K. Goebel, P. Khedkar, Hybrid soft computing systems: industrial and commercial applications, *Proc. IEEE* 87 (9) (1999) 1641–1667.
- [6] P. Bonissone, R. Subbu, N. Eklund, T. Kiehl, Evolutionary algorithms + domain knowledge = real-world evolutionary computation, *IEEE Trans. Evolutionary Comput.* 10 (2) (2006) 256–280.
- [7] O. Cerdón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, *Fuzzy Sets and Systems* 141 (1) (2004) 5–31.
- [8] C. Cruz Corona, Soft computing based multiagent cooperative strategies for solving optimization problems, Ph.D. Dissertation, University of Granada, 2005 (in Spanish).
- [9] F. Glover, Future paths for integer programming and links to artificial intelligence, *Comput. Oper. Res.* 13 (5) (1986) 533–549.
- [10] J.J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Trans. Systems Man Cybernet.* 16 (1) (1986) 122–128.
- [11] P. Larrañaga, J.A. Lozano, Synergies between evolutionary computation and probabilistic graphical models, *Internat. J. Approx. Reason.* 31 (3) (2002) 155–156.
- [12] P. Larrañaga, J.A. Lozano, H. Mühlenbein, Estimation of distribution algorithms in combinatorial optimization problems, *Rev. Iberoamericana Inteligencia Artificial* 19 (2) (2003) 149–168 (in Spanish).
- [13] X. Li, D. Ruan, A.J. van der Wal, Discussion on soft computing at FLINS'96, *Internat. J. Intelligent Systems* 13 (2–3) (1998) 287–300.
- [14] B. Melián, J.A. Moreno-Pérez, J.M. Moreno-Vega, Metaheuristics: a global view, *Rev. Iberoamericana Inteligencia Artificial* 19 (2) (2003) 7–28 (in Spanish).
- [15] I.H. Osman, G. Laporte, Metaheuristic: a bibliography, *Ann. Oper. Res.* 63 (1996) 513–623.
- [16] D.A. Pelta, Heuristics algorithms in bioinformatics, Ph.D. Dissertation, University of Granada, 2002 (in Spanish).
- [17] D.A. Pelta, A. Sancho-Royo, C. Cruz, J.L. Verdegay, Using memory and fuzzy rules in a co-operative multi-thread strategy for optimization, *Inform. Sci.* 176 (2006) 1849–1868.
- [18] J.L. Verdegay, (Ed.), *Fuzzy sets-Based Heuristics for Optimization. Studies in Fuzziness*, Springer, Berlin, 2003.
- [19] J.L. Verdegay, A revision of the methodologies involved in soft computing, in: Proc. I Symp. on Fuzzy Logic and Soft Computing (LFSC2005), Granada, Spain, 2005, pp. 151–156 (in Spanish).
- [20] J.L. Verdegay, E. Vergara-Moreno, Fuzzy termination criteria in knapsack problem algorithms, *Mathware and Soft Comput.* VII (2–3) (2000) 89–97.
- [21] E. Vergara-Moreno, New stop criteria in optimization algorithms, Ph.D. Dissertation, University of Granada, 1999 (in Spanish).
- [22] S. Voss, S. Martello, I.H. Osman, C. Rucairo (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Dordrecht, 1999.
- [23] L.A. Zadeh, Fuzzy sets. *Inform. and Control* (1965) 338–353.
- [24] L.A. Zadeh, Fuzzy logic and soft computing: issues, contentions and perspectives, in: Proc. IIZUKA'94: 3rd Internat. Conf. on Fuzzy Logic, Neural Nets and Soft Computing, Iizuka, Japan, 1994, pp. 1–2.
- [25] L.A. Zadeh, Soft computing and fuzzy logic, *IEEE Software* 11 (6) (1994) 48–56.
- [26] L.A. Zadeh, Applied soft computing, *Appl. Soft Comput.* 1 (2001) 1–2.