Chapter 1

Memetic Algorithms

1.1 Introduction

Back in the late 60s and early 70s, several researchers laid the foundations of what we now know as *evolutionary algorithms* [75, 108, 218, 227] (EAs). In these almost four decades, and despite some hard beginnings, most researchers interested in search or optimization –both from the applied and the theoretical standpoints– have grown to know and accept the existence –and indeed the usefulness– of these techniques. This has been also the case for other related techniques, such as *simulated annealing* [122] (SA), *tabu search* [83] (TS), etc. The name *metaheuristics* is used to collectively term these techniques.

It was in late 80s that the term *'Memetic Algorithms'* [178] (MAs) was given birth to denote a family of metaheuristics that tried to blend several concepts from tightly separated –at that time– families such as EAs and SA. The adjective 'memetic' comes from the term 'meme', coined by R. Dawkins [62] to denote an analogous to the *gene* in the context of cultural evolution. Quoting Dawkins:

> *"Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain via a process which, in the broad sense, can be called imitation."*

The above quote illustrates the central philosophy of MAs: individual improvement plus populational cooperation. As it was the case for classical EAs, MAs had to suffer tough initial times, but they are now becoming increasingly popular, as the reader may check by taking a quick look at the review of current work in MAs done at the end of this chapter. It is often the case that MAs are used under a different name ('hybrid EAs' and

'Lamarckian EAs' are two popular choices for this). Not quite surprisingly in a rapidly expanding field as this is, one can also find the term MA used in the context of particular algorithmic subclasses, arguably different from those grasped in the initial definition of MAs. This point will be tackled in next section; anticipating further definitions, we can say that a MA is a search strategy in which a population of optimizing agents synergistically cooperate and compete [189]. A more detailed description of the algorithm, as well as an functional template will be given in Section 1.2.

As mentioned before, MAs are a hot topic nowadays, mainly due to their success in solving many hard optimization problems. A particular feature of MAs is greatly responsible for this: unlike traditional Evolutionary Computation (EC) methods, MAs are intrinsically concerned with exploiting *all available knowledge* about the problem under study; this is something that was neglected in EAs for a long time, despite some contrary voices such as Hart and Belew [100], and most notably Davis [61]. The formulation of the so-called *No-Free-Lunch Theorem* (NFL) by Wolpert and Macready [247] made it definitely clear that a search algorithm strictly performs in accordance with the amount and quality of the problem knowledge they incorporate, thus backing up one of the *leiv motivs* of MAs.

The exploitation of problem-knowledge can be accomplished in MAs in a by incorporating heuristics, approximation algorithms, local search techniques, specialized recombination operators, truncated exact methods, etc. Also, an important factor is the use of adequate representations of the problem being tackled. These issues are of the foremost interest from an applied viewpoint, and will be dealt in Section 1.3.

As important as the basic algorithmic considerations about MAs that will be presented below, a more applied perspective of MAs is also provided in Section 1.4. The reader may be convinced of the wide applicability of these techniques by inspecting the numerous research papers published with regard to the deployment of MAs on the most diverse domains. We will pay special attention to the application of MAs in Engineering-related endeavors. This chapter will end with a brief summary of the current research trends in MAs, with special mention to those emerging application fields in which MAs are to play a major rôle in the near future.

1.2 The MA Search Template

As mentioned in the previous section, MAs try to blend together concepts from different metaheuristics, such as EAs and SA for instance. Let us start by those ideas gleaned from the former.

MAs are –like EAs– population-based metaheuristics. This means that the algorithm maintain a *population* of solutions for the problem at hand, i.e., a pool comprising several solutions simultaneously. Each of these solutions is termed *individual* in the EA jargon, following the nature-inspired metaphor upon which these techniques are based. In the context of MAs, the denomination *agent* seems more appropriate for reasons that will be evident later in this section. When clear from the context, both terms will be used interchangeably.

Each individual –or agent– represents a tentative solution for the problem under consideration. These solutions are subject to processes of competition and mutual cooperation in a way that resembles the behavioral patterns of living beings from a same species. To make clearer this point, it is firstly necessary to consider the high-level template of the basic populational event: a *generation*. This is shown below in Fig. 1.1.

Process Do-Generation ($\downarrow\uparrow$ *pop* : *individual*[ ])
variables
  $breeders, newpop : Individual[\,]$;
begin
  breeders ← Select-From-Population(*pop*);
  newpop ← Generate-New-Population(*breeders*);
  pop ← Update-Population (*pop*, *newpop*)
end

Figure 1.1: The basic generational step

As it can be seen, each generation consists of the updating of a population of individuals, hopefully leading to better and better solutions for the problem being tackled. There are three main components in this generational step: selection, reproduction, and replacement. The first component (selection) is responsible (jointly with the replacement stage) for the competition aspects of individuals in the population. Using the information provided by an *ad hoc* guiding function (*fitness* function in the EA terminology), the goodness of individuals in *pop* is evaluated; subsequently, a sample of individuals is selected for reproduction according to this goodness measure. This selection can be done in a variety of ways. The most popular techniques are fitness-proportionate methods (the probability of selecting an individual for breeding is proportional to its fitness[1]), rank-based methods (the probability of selecting an individual depends on its position after ranking the whole population), and tournament-based methods (individuals are selected on the basis of a direct competition within small sub-groups of individuals).

Replacement is very related to this competition aspect, as mentioned above. This component takes care of maintaining the population at a con-

[1]Maximization is assumed here. In case we were dealing with a minimization problem, fitness should be transformed so as to obtain an appropriate value for this purpose, e.g., subtracting it from the highest possible value of the guiding function

stant size. To do so, individuals in the older population are substituted by the newly-created ones (obtained from the reproduction stage) using some specific criterion. Typically, this can be done by taking the best (according to the guiding function) individuals both from *pop* and *newpop* (the so-called "plus" replacement strategy), or by simply taking the best individuals from *newpop* and inserting them in *pop* substituting the worst ones (the "comma" strategy). In the former case, if $|pop| = |newpop|$ then the replacement is termed *generational*; if $|newpop|$ is small (say $|newpop| = 1$), then we have a *steady-state* replacement.

Maybe the most interesting aspect in this generation process is the intermediate phase of reproduction. At this stage, we have to create new individuals (or agents) by using the existing ones. This is done by utilizing a number of reproductive *operators*. Many different such operators can be used in a MA, as illustrated in the general pseudocode shown in Fig. 1.2. Nevertheless, the most typical situation involves utilizing just two operators: recombination and mutation.

Process Generate-New-Population
$(\downarrow pop : Individual[\,], \downarrow op : Operator[\,]) \rightarrow Individual[\,]$
variables
  $buffer : Individual[\,][\,]$;
  $j : [1..|op|]$;
begin
  $buffer[0] \leftarrow pop$;
  for $j \leftarrow 1{:}|op|$ **do**
    $buffer[j] \leftarrow$ Apply-Operator $(op[j], buffer[j-1])$;
  endfor;
  return $buffer[n_{op}]$
end

Figure 1.2: Generating the new population.

Recombination is a process that encapsulates the mutual cooperation among several individuals (typically two of them, but a higher number is possible [72]). This is done by constructing new individuals using the information contained in a number of selected *parents*. If it is the case that the resulting individuals (the *offspring*) are entirely composed of information taken from the parents, then the recombination is said to be *transmitting* [211]. This is the case of classical recombination operators for bitstrings such as *single-point crossover*, or *uniform crossover* [233]. This property captures the *a priori* rôle of recombination as previously enunciated, but it can be difficult to achieve for certain problem domains (the *Traveling Salesman Problem* –TSP– is a typical example). In those situations, it is possible to consider other properties of interest such as *respect* or *assortment*. The former refers to the fact that the recombination operator generate descendants carrying all *features* (i.e., basic properties of solutions with relevance for the problem attacked) common to all parents; thus, this property can be seen as a part of the *exploitative* side of the search. On the other hand, *assortment* represents the exploratory side of recombination. A recombination operator is said to be *properly assorting* if, and only if, it can generate descendants carrying any combination of compatible features taken from the parents. The assortment is said to be *weak* if it is necessary to perform several recombinations within the offspring to achieve this effect.

Several interesting concepts have been introduced in this description of recombination, namely, *relevant features* and *cooperation*. We will return to these points in the next section. Before that, let us consider the other operator mentioned above: mutation. From a classical point of view (at least in the genetic-algorithm arena [84]), this is a secondary operator whose mission is to *keep to pot boiling*, continuously injecting new material in the population, but at a low rate (otherwise the search would degrade to a random walk in the solution space). Evolutionary-programming practitioners [75] would disagree with this characterization, claiming a central rôle for mutation. Actually, it is considered the crucial part of the search engine in this context.

In essence, a mutation operator must generate a new solution by partly modifying an existing solution. This modification can be random –as it is typically the case– or can be endowed with problem-dependent information so as to bias the search to probably-good regions of the search space. It is precisely in the light of this latter possibility that one of the most distinctive components of MAs is introduced: *local-improvers*. To understand their philosophy, let us consider the following abstract formulation: first of all, assume a mutation operator that performs a random minimal modification in a solution; now consider the graph whose vertices are solutions, and whose edges connect pairs of vertices such that the corresponding solutions can be obtained via the application of the mutation operator on one of them[2]. A local-improver is a process that starts at a certain vertex, and moves to an adjacent vertex, provided that the neighboring solution is better that the current solution. This is illustrated in Fig. 1.3.

As it can be seen, the local-improver tries to find an "uphill" (in terms of improving the value provided by the guiding function F_g) path in the graph whose definition was sketched before. The formal name for this graph is *fitness landscape* [115]. Notice that the length of the path found by the local-improver is determined by means of a Local-Improver-Termination-Criterion function. A usual example is terminating the path when no more uphill movements are possible (i.e., when the current solution is a local

[2]Typically this graph is symmetrical, but in principle there is no problem in assuming it to be asymmetrical.

Process Local-Improver ($\downarrow\uparrow$ $current$: $Individual$, $\downarrow$ op : $Operator$)
variables
  new : $Individual$
begin
  repeat
    $new \leftarrow$ Apply-Operator(op, $current$);
    if ($F_g(new) \prec_{\mathcal{F}} F_g(current)$) **then**
      $current \leftarrow new$;
    endif
  until Local-Improver-Termination-Criterion();
  return $current$;
end

Figure 1.3: Pseudocode of a Local-Improver

optimum with respect to op). However, this is not necessarily the case always. For instance, the path can be given a maximum allowed length, or it can be terminated as soon as the improvement in the value of the guiding function is considered good enough. For this reason, MAs cannot be characterized as *"EAs working in the space of local-optima [with respect to a certain fitness landscape]"*; that would be an unnecessarily restricted definition.

The local-improver algorithm can be used in different parts of the generation process, for it is nothing else than just another operator. For example, it can be inserted after the utilization of any other recombination or mutation operator; alternatively, it could be just used at the end of the reproductive stage. See [**?**] for examples of these settings.

As said before, the utilization of this local-improver[3] is one of the most characteristic features of MAs. It is precisely because of the use of this mechanism for improving individuals on a local (and even autonomous) basis that the term 'agent' is deserved. Thus, the MA can be viewed as a collection of agents performing an autonomous exploration of the search space, cooperating some times via recombination, and competing for computational resources due to the use of selection/replacement mechanisms.

After having presented the innards of the generation process, we can now have access to the larger picture. The functioning of a MA consists of the iteration of this basic generational step, as shown in Fig. 1.4.

Several comments must be made with respect to this general template. First of all, the Generate-Initial-Population process is responsible for creating the initial set of $|pop|$ configurations. This can be done by simply

[3]We use the term in singular, but notice that several different local-improvers could be used in different points of the algorithm.

Process MA () $\rightarrow$ *Individual*[ ]
variables
  pop : *Individual*[ ];
begin
  pop $\leftarrow$ Generate-Initial-Population();
  repeat
    pop $\leftarrow$ Do-Generation (*pop*)
    if Converged(*pop*) **then**
      pop $\leftarrow$ Restart-Population(*pop*);
    endif
  until MA-Termination-Criterion()
end

Figure 1.4: The general template of a MA

generating $|pop|$ random configurations or by using a more sophisticated seeding mechanism (for instance, some constructive heuristic), by means of which high-quality configurations are injected in the initial population [232] [147]. Another possibility, the Local-Improver presented before could be used as shown in Fig. 1.5:

Process Generate-Initial-Population ($\downarrow \mu : \mathbb{N}$) $\rightarrow$ *Individual*[ ]
variables
  pop : *Individual*[ ];
  ind : *Individual*;
  j : $[1..\mu]$;
begin
  for $j \leftarrow 1{:}\mu$ **do**
    ind $\leftarrow$ Generate-Random-Solution();
    pop[j] $\leftarrow$ Local-Improver (*ind*);
  endfor
  return *pop*
end

Figure 1.5: Injecting high-quality solutions in the initial population.

There is another interesting element in the pseudocode shown in Fig. 1.4: the Restart-Population process. This process is very important in order to make an appropriate use of the computational resources. Consider that the population may reach a state in which the generation of new improved solution be very unlikely. This could be the case when all agents in the population are very similar to each other. In this situation, the algorithm will probably expend most of the time resampling points in a very limited region of the search space [48], with the subsequent waste of computational efforts. This phenomenon is known as *convergence*, and it can be identified using measures such as Shannon's entropy [60]. If this measure falls below a predefined threshold, the population is considered at a degenerate state. This threshold depends upon the representation of the problem being used (number of values per variable, constraints, etc.) and hence must be determined in an *ad-hoc* fashion. A different possibility is using a probabilistic approach to determine with a desired confidence that the population has converged. For example, in [111] a Bayesian approach is presented for this purpose.

Once the population is considered to be at a degenerate state, the restart process is invoked. Again, this can be implemented in a number of ways. A very typical strategy is keeping a fraction of the current population (this fraction can be as small as one solution, the current best), and substituting the remaining configurations with newly generated (from scratch) solutions, as shown in Fig. 1.6:

Process Restart-Population ($\downarrow pop : Individual[\,]$)$\rightarrow Individual[\,]$
variables
  $newpop : Individual[\,]$;
  $j, \#preserved : [1..|pop|]$;
begin
  $\#preserved \leftarrow |pop| \cdot \%PRESERVE$;
  for $j \leftarrow 1{:}\#preserved$ **do**
    $newpop[j] \leftarrow$ ithBest(pop, j);
  endfor
  for $j \leftarrow (\#preserved + 1) : |pop|$ **do**
    $newpop[j] \leftarrow$ Generate-Random-Configuration();
    $newpop[j] \leftarrow$ Local-Improver ($newpop[j]$);
  endfor;
  return *newpop*
end

Figure 1.6: A possible re-starting procedure for the population.

The above process completes the functional description of MAs. Obviously, it is possible to conceive some *ad-hoc* modifications of this template that still could be catalogued as MA. The reader can nevertheless be ensured that any such algorithm will follow the general philosophy depicted in this section, and could be possibly rewritten so as to match this template.

1.3 Design of Effective MAs

The general template of MAs we have depicted in the previous section must be instantiated with precise components in order to be used for solving an specific problem. This instantiation has to be done carefully so as to obtain an effective optimization tool. We will address some design issues in this section.

A first obvious remark must be done: there exist no general approach for the design of effective MAs. This fact admits different proofs depending on the precise definition of *effective* in the previous statement. Such proofs may involve classical complexity results and conjectures if 'effective' is understood as 'polynomial-time', the NFL Theorem if we consider a more general set of performance measures, and even Computability Theory if we relax the definition to arbitrary decision problems. For these reasons, we can only define several *design heuristics* that will likely result in good-performing MAs, but without explicit guarantees for this.

Having introduced this point of caution, the first element that one has to decide is the *representation* of solutions. At this point it is necessary to introduce a subtle but important distinction here: representation and *codification* are different things. The latter refers to the way solutions are internally stored, and it can be chosen according to memory limitations, manipulation complexity, and other resource-based considerations. On the contrary, the representation refers to an abstract formulation of solutions, relevant from the point of view of the functioning of reproductive operators. This duality was present in discussions contemporary to the early debate on MAs (e.g., see [210]), and can be very well-exemplified in the context of permutational problems. For instance, consider the TSP; solutions can be internally encoded as permutations, but if a edge-recombination operator is used (e.g., [150]) then solutions are *de facto* represented as edge lists.

The above example about the TSP also serves for illustrating one of the properties of representations that must be sought. Consider that a permutation can be expressed using different *information units*; for instance, it can be determined on the basis of the specific values of each position. This is the *position-based* representation of permutations [84]. On the other hand, it can be determined on the basis of adjacency relationships between the elements of the permutation. Since the TSP is defined by a matrix of inter-city distances, it seems that edges are more relevant for this problem than absolute positions in the permutation. In effect, it turns out that operators manipulating this latter representation perform better than operators that manipulate positions such as *partially-mapped crossover* [85] (PMX) or *cycle crossover* [191] (CX).

There have been several attempts for quantifying how good a certain set of information units is for representing solutions for a specific problems. We can cite a few of them:

- *Minimizing epistasis*: epistasis can be defined as the non-additive influence on the guiding function of combining several information units (see [59] for example). Clearly, the higher this non-additive influence, the lower the absolute relevance of individual information units. Since the algorithm will be processing such individual units (or small groups of them), the guiding function turns out to be low informative, and prone to misguide the search.

- *Minimizing fitness variance* [212]: This criterion is strongly related to the previous one. The fitness variance for a certain information unit is the variance of the values returned by the guiding function, measured across a representative subset of solutions carrying this information unit. By minimizing this fitness variance, the information provided by the guiding function is less *noisy*, with the subsequent advantages for the guidance of the algorithm.

- *Maximizing fitness correlation*: In this case a certain reproductive operator is assumed, and the correlation in the values of the guiding function for parents and offspring is measured. If the fitness correlation is high, good solutions are likely to produce good solutions, and thus the search will gradually shift toward the most promising regions of the search space. Again, there is a clear relationship with the previous approaches; for instance, if epistasis (or fitness variance) is low, then solutions carrying specific features will have similar values for the guiding function; since the reproductive operators will create new solutions by manipulating these features, the offspring is likely to have a similar guiding value as well.

Obviously, the description of these approaches may appear somewhat idealized, but the underlying philosophy is well illustrated. It must be noted that selecting a representation is not an isolated process, but it has a strong liaison with the task of choosing appropriate reproductive operators for the MA. Actually, according to the operator-based view of representations described above, the existence of multiple operators may imply the consideration of different representations of the problem at different stages of the reproductive phase. We will come back to this issue later in this section.

In order to tackle the operator-selection problem, we can resort to existing operators, or design new *ad hoc* operators. In the former case, a suggested line of action could be the following [49]:

1. We start from a set of existing operators $\Omega = \{\omega_1, \omega_2, \cdots, \omega_k\}$. The first step is identifying the representation of the problem manipulated by each of these operators.

2. Use any of the criterions presented for measuring the goodness of the representation.

3. Select ω_i from Ω, such that the representation manipulated by ω_i is the more trustable.

This is called *inverse analysis of operators* since some kind of inverse engineering is done in order to evaluate the potential usefulness of each operator. The alternative would be a *direct analysis* in which new operators would be designed. This could be do as follows:

1. Identify different potential representation for the problem at hand (e.g., recall the previous example on the TSP).

2. Use any of the criterions presented for measuring the goodness of these representation.

3. Create new operators $\Omega' = \{\omega'_1, \omega'_2, \cdots, \omega'_m\}$ via the manipulation of the most trustable information units.

In order to accomplish the last step of the *direct analysis*, there exists a number of templates for the manipulation of abstract information units. For example, the templates known as *random respectful recombination* (R^3), *Random Assorting Recombination* (RAR), and *Random Transmitting Recombination* (RTR) have been defined in [211]. An example of the successful instantiation of some of these templates using the direct analysis in the context of flowshop scheduling can be found in [52].

The generic templates mentioned above are essentially *blind*. This means that they do not use problem-dependent information at any stage of their functioning. This use of blind recombination operators is traditionally justified on the grounds of not introducing excessive bias in the search algorithm, thus preventing extremely fast convergence to suboptimal solutions. However, this is a highly arguable point since the behavior of the algorithm is in fact biased by the choice of representation. Even if we neglect this fact, it can be reasonable to pose the possibility of quickly obtaining a suboptimal solution and restarting the algorithm, rather than using blind operators for a long time in pursuit of an asymptotically optimal behavior (not even guaranteed in most cases).

Reproductive operators that use problem knowledge are commonly termed *heuristic* or *hybrid*. In these operators, problem information is utilized to guide the process of producing the offspring. There are numerous ways to achieve this inclusion of problem knowledge; in essence, we can identify two major aspects into which problem knowledge can be injected: the selection of the parental features that will be transmitted to the descendant, and the selection of non-parental features that will be added to it[4].

[4]Notice that the use of the term 'parental information' does not imply the existence of more than one parent. In other words, the discussion is not restricted to recombination operators, but may also include mutation operators.

With respect to the selection of parental features to be injected in the offspring, there exists evidence that *respect* (transmission of common features, as mentioned in the previous section) is beneficial for some problems (e.g., see [51][150]). After this initial transmission, the offspring can be completed in several ways. For example, Radcliffe and Surry [212] have proposed the use of local-improvers or implicit enumeration schemas[5]. This is done by firstly generating a partial solution by means of a non-heuristic procedure; subsequently, two approaches can be used:

- *locally-optimal completion*: the child is completed at random, and a local-improver is used restricted to those information units added for completion.

- *globally-optimal completion*: an implicit enumeration schema is used in order to find the globally best combination of information units that can be used to complete the child.

Related to the latter approach, the implicit enumeration schema can be used to find the best combination of the information units present in the parents. The resulting recombination would thus be *transmitting*, but not necessarily *respectful* for these two properties are incompatible in general. However, respect can be enforced by restricting the search to non-common features. Notice that this would not be globally-optimal completion since the whole search is restricted to information comprised in the parents. The set of solutions that can be constructed using this parental information is termed *dynastic potential*, and for this reason this approach is termed *dynastically optimal recombination* [56] (DOR). This operator is monotonic in the sense that any child generated is at least as good as the best parent.

Problem-knowledge need not be necessarily included via iterative algorithms. On the contrary, the use of constructive heuristics is a popular choice. A distinguished example is the *Edge Assembly Crossover* (EAX) [186]. EAX is a specialized operator for the TSP (both for symmetric and asymmetric instances) in which the construction of the child comprises two-phases: the first one involves the generation of an incomplete child via the so-called E-sets (subtours composed of alternating edges from each parent); subsequently, these subtours are merged into a single feasible subtours using a greedy repair algorithm. The authors of this operator reported impressive results in terms of accuracy and speed. It has some similarities with the recombination operator proposed in [179].

To some extent, the above discussion is also applicable to mutation operators, although these exhibit a clearly different rôle: they must introduce new information. This means that purely transmitting mechanisms would

[5]Actually, these approaches can be used even when no initial transmission of common features is performed.

not be acceptable for this purpose. Nevertheless, it is still possible to use the ideas described in the previous paragraphs by noting that the 'partial solution' mentioned in several situations can be obtained by simply removing some information units from a single solution. A completion procedure as described before can then be used in order to obtain the mutated solution.

Once we have one or more knowledge-augmented reproductive operators, it is necessary to make them work in a synergistic fashion. This is a feature of MAs that is also exhibited by other metaheuristics such as *variable neighborhood search* (VNS) [98], although it must me emphasized that it was already included in the early discussions of MAs, before the VNS metaheuristic was formulated. We can quote from [177]:

> *"Another advantage that can be exploited is that the most powerful computers in the network can be doing the most time-consuming heuristics, while others are using a different heuristics. The program to do local search in each individual can be different. This enriches the whole, since what is a* local minima *for one of the computers is not a local minima for another in the network. Different heuristics may be working fine due to different reasons. The collective use of them would improve the final output. In a distributed implementation we can think in a division of jobs, dividing the kind of moves performed in each computing individual. It leads to an interesting concept, where instead of dividing the physical problem (assignment of cities/cells to processors) we divide the set of possible moves. This set is selected among the most efficient moves for the problem."*

This idea of synergistically combining different operators (and indeed different search techniques) was exemplified at its best by Applegate, Bixby, Cook, and Chvatal in 1998. They established new breakthrough results for the Min TSP which supports our view that MAs will have a central role as a problem solving methodology. This team solved to optimality an instance of the TSP of 13,509 cities corresponding to all U.S. cities with populations of more than 500 people [6]. The approach, according to Bixby: *"...involves ideas from polyhedral combinatorics and combinatorial optimization, integer and linear programming, computer science data structures and algorithms, parallel computing, software engineering, numerical analysis, graph theory, and more"*. Their approach can possibly be classified as the most complex MA ever built for a given combinatorial optimization problem.

These ideas have been further developed in a recent unpublished manuscript, *"Finding Tours in the TSP"* by the same authors (Bixby *et al.*), available from their web site. They present results on running an optimal algorithm for solving the Min Weighted Hamiltonian Cycle Problem in a sub-

[6]See: `http://www.crpc.rice.edu/CRPC/newsArchive/tsp.html`

graph formed by the union of 25 Chained Lin-Kernighan tours. The approach consistently finds the optimal solution to the original Min TSP instances with up to 4461 cities. They also attempted to apply this idea to an instance with 85,900 cities (the largest instance in TSPLIB) and from that experience they convinced themselves that it also works well for such large instances.

The approach of running a local search algorithm (Chained Lin Kernighan) to produce a collection of tours, following by the dynastical-optimal recombination method the authors named *tour merging* gave a non-optimal tour of only 0.0002 % excess above the proved optimal tour for the 13,509 cities instance. We take this as a clear proof of the benefits of the MA approach and that more work is needed in developing good strategies for *complete memetic algorithms*, i.e., those that systematically and synergistically use randomized and deterministic methods and can prove optimality.

We would like to close this section by emphasizing once again the heuristic nature of the design principles described in this section. The most interesting thing to note here is not the fact that they are just probably-good principles, but the fact that there is still much room for research in methodological aspects of MAs (e.g., see [125]). The open-philosophy of MAs make them suitable for incorporating mechanisms from other optimization techniques. In this sense, the reader may find a plethora of new possibilities for MA design by studying other metaheuristics such as TS, for example.

1.4 Applications of MAs

This section will provide an overview of the numerous applications of MAs. This overview is far from exhaustive since new applications are being developed continuously. However, it is intended to be illustrative of the practical impact of these optimization techniques.

1.4.1 NP-hard Combinatorial Optimization problems

Traditional NP Optimization problems constitute one of the most typical battlefields of MAs. A remarkable history of successes has been reported with respect to the application of MAs to $NP-$hard problems such as the following: Graph Partitioning [21] [22] [159] [162] [163], Min Number Partitioning [16] [17], Max Independent Set [3] [102] [225], Bin-Packing [219], Min Graph Coloring [44] [47] [70] [74], Set Covering [12], Min Generalised Assignment [41], Multidimensional Knapsack [13] [53] [91], Nonlinear Integer Programming [234], Quadratic Assignment [20] [35] [157] [161] [162], Quadratic Programming [164][166], Set Partitioning [138], and particularly on the Min Travelling Salesman Problem and its variants [79] [78] [88] [89] [90] [109] [119] [128] [156] [158] [162] [165] [181] [213] [222] .

Regarding the theory of NP-Completeness, most of them can be cited as *"classical"* as they appeared in Karp's notorious paper [117] on the reducibility of combinatorial problems. Remarkably, in most of them the authors claim that they have developed the best heuristic for the problem at hand. This is important since these problems have been addressed with several with different approaches from the combinatorial optimization toolbox and almost all general-purpose algorithmic techniques have been tested on them.

The MA paradigm is not limited to the above mentioned classical problems. There exist additional "non-classical" combinatorial optimization problems of similar or higher complexity in whose resolution MAs have revealed themselves as outstanding techniques. As an example of these problems, one can cite *partial shape matching* [196], *Kauffman NK Landscapes* [160], *spacecraft trajectory design* [57], *minimum weighted k-cardinality tree subgraph problem* [18], *minimum k-cut problem* [251], *uncapacitated hub location* [2], *placement problems* [110] [134] [226], *vehicle routing* [15] [113], *transportation problems* [82] [190], and *task allocation* [97].

Another important class of combinatorial optimization problems are those that directly or indirectly correspond to telecommunication network problems. For example, we can cite: *frequency allocation* [55] [118], *network design* [81] [224], *degree-constrained minimum spanning tree problem* [214], *vertex-biconnectivity augmentation* [120], *assignment of cells to switches in cellular mobile networks* [209], and *OSPF routing* [23],

Obviously, this list is by no means complete since its purpose is simply to document the wide applicability of the approach for combinatorial optimization.

1.4.2 Scheduling Problems

Undoubtedly, scheduling problems are one of the most important optimization domains due to its practical implications. They thus deserve separate mention, despite they could be included in the NP-hard class surveyed in the previous subsection.

MAs have been used to tackle a large variety of scheduling problems. We can cite the following: *maintenance scheduling* [28] [29] [30], *open shop scheduling* [40] [73] [142], *flowshop scheduling* [10] [36] [183] [184], *total tardiness single machine scheduling* [153], *single machine scheduling with setup-times and due-dates* [76] [137] [170], *parallel machine scheduling* [38] [39] [154] [172], *project scheduling* [188] [197] [215], *warehouse scheduling* [240], *production planning* [67] [173], *timetabling* [24] [25] [26] [27] [31] [87] [145] [175] [176] [200] [201] [216], *rostering* [63] [174], and *sport games scheduling* [46].

1.4.3 Machine Learning and Robotics

Machine learning and robotics are two closely related fields since the different tasks involved in the control of robots are commonly approached using artificial neural networks and/or classifier systems. MAs, generally cited as "genetic hybrids" have been used in both fields, i.e., in general optimization problems related to machine learning (for example, the training of artificial neural networks), and in robotic applications. With respect to the former, MAs have been applied to *neural network training* [1] [112] [179] [236] [249], *pattern recognition* [4], *pattern classification* [132] [169], and *analysis of time series* [71] [193].

As to the application of MAs to robotics, work has been done in *reactive rulebase learning in mobile agents* [54], *path planning* [192] [205] [248], *manipulator motion planning* [221], *time optimal control* [37], etc.

1.4.4 Engineering, Electronics and Electromagnetics

Electronics and engineering are also two fields in which these methods have been actively used. For example, with regard to engineering problems, work has been done in the following areas: *structure optimization* [250], *system modeling* [239], *fracture mechanics* [198], *aeronautic design* [19] [208], *trim loss minimization* [194], *traffic control* [231], *power planning* [237], *calibration of combustion engines* [123] [204], and *process control* [45] [254].

As to practical applications in the field of electronics and electromagnetics [42], the following list can illustrate the numerous areas in which these techniques have been utilized: *semiconductor manufacturing* [121], *circuit design* [6] [7] [94] [99] [244], *circuit partitioning* [5] *computer aided design* [14], *multilayered periodic strip grating* [9], *analogue network synthesis* [92], *service restoration* [8], *optical coating design* [107], and *microwave imaging* [33] [203].

1.4.5 Molecular Optimization Problems

We have selected this particular class of computational problems, involving nonlinear optimization issues, to help the reader to identify a common trend in the literature. Unfortunately, the authors continue referring to their technique as 'genetic', although they are closer in spirit to MAs [106].

The Caltech report that gave its name to the, at that time incipient, field of MAs [177] discussed a metaheuristic which can be viewed as a hybrid of GAs and SA developed with M.G. Norman in 1988. In recent years, several papers applied hybrids of GAs with SA or other methods to a variety of molecular optimization problems [11] [58] [64] [69] [80] [93] [114] [116] [135] [140] [146] [148] [171] [155] [199] [228] [229] [235] [245] [253] [255]. Hybrid population approaches like this can hardly be catalogued as being 'genetic', but this denomination has appeared in previous work by Deaven and Ho [65] and then cited by J. Maddox in *Nature* [149]. Other fields of application include *cluster physics* [187]. Additional work has been done in [66] [104] [105] [206] [207] [245]. Other evolutionary approaches to a variety of molecular problems can be found in: [69] [101] [103] [152] [168] [217] [238]. Their use for design problems is particularly appealing [43] [116] [246]. They have also been applied in protein design [68] [136], structure prediction [126] [127] [131], and alignment [34] (see also the discussion in [179] and the literature review in [106]).

This field is enormously active, and new application domains for MAs are continuously emerging. Among these, we must mention applications related to genomic analysis, such as *clustering gene-expression profiles* [167], or *inferring phylogenetic trees* [50].

1.4.6 Other Applications

In addition to the application areas described above, MAs have been also utilized in other fields such as, for example, *medicine* [95] [96] [241], *economics* [139] [195], *oceanography* [185], *mathematics* [220] [242] [243], *imaging science and speech processing* [32] [133] [141] [151] [223] [252], etc.

For further information about MA applications we suggest querying bibliographical databases or web browsers for the keywords *'memetic algorithms'* and *'hybrid genetic algorithm'*. We have tried to be illustrative rather than exhaustive, pointing out some selected references for well-known application areas. This means that, with high probability, many important contributions may have been inadvertently left out.

1.5 Conclusions and Future Directions

We believe that MAs have very favorable perspectives for their development and widespread application. Such a belief is grounded on several reasons. First of all, MAs are showing a great record of efficient implementations, providing very good results in practical problems as the reader may have checked by inspecting the previous section. We also have reasons to believe that we are near some major leaps forward in our theoretical understanding of these techniques, including for example the worst-case and average-case computational complexity of recombination procedures. On the other hand, the ubiquitous nature of distributed systems, like networks of workstations for example, plus the inherent asynchronous parallelism of MAs and the existence of web-conscious languages like Java are all together an excellent combination to develop highly portable and extendable object-oriented frameworks allowing algorithmic reuse.

We also see as a healthy sign the systematic development of other particular optimization strategies. If any of the simpler metaheuristics (SA, TS, VNS, GRASP, etc.) performs the same as a more complex method (GAs, MAs, Ant Colonies, etc.), an "elegance design" principle should prevail and we must either resort to the simpler method, or to the one that has less free parameters, or to the one that is easier to implement. Such a fact should defy us to adapt the complex methodology to beat a simpler heuristic, or to check if that is possible at all. An unhealthy sign of current research, however, are the attempts to encapsulate metaheuristics on stretched confinements.

We think that there are several "learned lessons" from work in other metaheuristics. For instance, a Basic Tabu Search scheme ([83]) decides to accept another new configuration (whether a feasible solution or not) without restriction to the relative objective function value of the two solutions. This has lead to good performance in some configuration spaces where evolutionary methods and Simulated Annealing perform poorly. A classical example of this situation is the Min Number Partitioning problem [17].

There are many open lines of research in areas such as co-evolution. In [180] we can find the following quotation:

> *"It may be possible that a future generation of MAs will work in at least two levels and two time scales. In the short-time scale, a set of agents would be searching in the search space associated to the problem while the long-time scale adapts the heuristics associated with the agents. Our work with D. Holstein which will be presented in this book might be classified as a first step in this promising direction. However, it is reasonable to think that more complex schemes evolving solutions, agents, as well as representations, will soon be implemented."*

At that time, we were referring to the use of a metaheuristic called *Guided Local Search* used in [109] as well as the possibility of co-evolving the neighborhood techniques by other means. Unfortunately, this was not studied in depth in Holstein's thesis [**?**]. However, a number of more recent articles are paving the way to more robust MAs [34, 124, 129, 130]. Krasnogor has recently introduced the term *multimeme* algorithms to identify those MAs that also adaptively change the neighborhood definition [131], and with colleagues is applying the method for the difficult problem of *protein structure prediction* [127]. Smith also presents a recent study on these issues in [230].

More work is necessary, and indeed the protein folding models they are using are a good test-bed for the approach. However, we also hope that the researchers should again concentrate MAs for large-scale challenging instances of the TSP, possibly following the approaches of using population structures [182, 77], self-adapting local search, [128] as well as the powerful recombination operators that have been devised for TSP instances [109, 156, 165, 179]. We have also identified some problems with evolutionary search methods in instances of the TSP in which the entries of the distance matrix have a large number of decimal digits. This means that there is an inherent problem to be solved, for evolutionary methods to deal with fitness functions that have so many decimal digits. Traditional rank-based or fitness-based selection schemes to keep new solutions in the current population fail. It would be then reasonable to investigate whether some ideas from basic TS mechanisms could be adapted to allow less stringent selection approaches.

Multiparent recombination is also an exciting area to which research efforts can be directe4d too. From [202] we can read:

> *"The strategy developed by Lin [143] for the TSP is to obtain several local optima and then identify edges that are common to all of them. These are then fixed, thus reducing the time to find more local optima. This idea is developed further in [144] and [86]."* It is intriguing that such an strategy, which has been around for more than three decades, is still not accepted by some researchers.

We think that the use of *multiparent* recombination with proven good properties is one of the most challenging issues for future developement in MAs, as well as for the whole EC paradigm.

Bibliography

[1] Abbass, H. A memetic Pareto evolutionary approach to artificial neural networks. *Lecture Notes in Computer Science 2256* (2001), 1–??

[2] Abdinnour, H. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operational Research 106*, 2-3 (1998), 489–99.

[3] Aggarwal, C., Orlin, J., and Tai, R. Optimized crossover for the independent set problem. *Operations Research 45*, 2 (1997), 226–234.

[4] Aguilar, J., and Colmenares, A. Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. *Pattern Analysis and Applications 1*, 1 (1998), 52–61.

[5] Areibi, S. An integrated genetic algorithm with dynamic hill climbing for VLSI circuit partitioning. In *Data Mining with Evolutionary Algorithms* (Las Vegas, Nevada, USA, 8 July 2000), A. A. Freitas, W. Hart, N. Krasnogor, and J. Smith, Eds., pp. 97–102.

[6] Areibi, S. Memetic algorithms for VLSI physical design: Implementation issues. In *Second Workshop on Memetic Algorithms (2nd WOMA)* (San Francisco, California, USA, 7 July 2001), W. Hart, N. Krasnogor, and J. Smith, Eds., pp. 140–145.

[7] Areibi, S. The performance of memetic algorithms on physical design, March 2002. Submitted to *Journal of Applied Systems Studies*, Special Issue: Real Life Applications of Nature Inspired Combinatorial Heuristics.

[8] Augugliaro, A., Dusonchet, L., and Riva-Sanseverino, E. Service restoration in compensated distribution networks using a hybrid genetic algorithm. *Electric Power Systems Research 46*, 1 (1998), 59–66.

[9] Aygun, K., Weile, D., and Michielssen, E. Design of multilayered periodic strip gratings by genetic algorithms. *Microwave and Optical Technology Letters 14*, 2 (1997), 81–85.

[10] Basseur, M., Seynhaeve, F., and Talbi, E. Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation, CEC'02, May 12-17, 2002, Honolulu, Hawaii, USA* (2002), X. Yao, Ed., pp. 1151–1156.

[11] Bayley, M., Jones, G., Willett, P., and Williamson, M. Genfold: A genetic algorithm for folding protein structures using NMR restraints. *Protein Science 7*, 2 (1998), 491–499.

[12] Beasley, J., and Chu, P. A genetic algorithm for the set covering problem. *European Journal of Operational Research 94*, 2 (1996), 393–404.

[13] Beasley, J., and Chu, P. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics 4* (1998), 63–86.

[14] Becker, B., and Drechsler, R. Ofdd based minimization of fixed polarity Reed-Muller expressions using hybrid genetic algorithms. In *Proceedings IEEE International Conference on Computer Design: VLSI in Computers and Processor* (Los Alamitos, CA, 1994), IEEE, pp. 106–110.

[15] Berger, J., Salois, M., and Begin, R. A hybrid genetic algorithm for the vehicle routing problem with time windows. In *Advances in Artificial Intelligence. 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence* (Berlin, 1998), R. Mercer and E. Neufeld, Eds., Springer-Verlag, pp. 114–127.

[16] Berretta, R., Cotta, C., and Moscato, P. Forma analysis and new heuristic ideas for the number partitioning problem. In *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001* (2001), J. P. de Sousa, Ed., pp. 337–341.

[17] Berretta, R., and Moscato, P. The number partitioning problem: An open challenge for evolutionary computation ? In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999, pp. 261–278.

[18] Blesa, M., Moscato, P., and Xhafa, F. A memetic algorithm for the minimum weighted k-cardinality tree subgraph problem. In *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001* (2001), J. P. de Sousa, Ed., pp. 85–90.

[19] Bos, A. Aircraft conceptual design by genetic/gradient-guided optimization. *Engineering Applications of Artificial Intelligence 11*, 3 (1998), 377–382.

[20] Brown, D., Huntley, C., and Spillane, A. A Parallel Genetic Heuristic for the Quadratic Assignment Problem. In *Proceedings of the Third International Conference on Genetic Algorithms* (1989), J. Schaffer, Ed., Morgan Kaufmann, pp. 406–415.

[21] Bui, T., and Moon, B. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers 45*, 7 (1996), 841–855.

[22] Bui, T., and Moon, B. GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 17*, 3 (1998), 193–204.

[23] Buriol, L., Resende, M., Ribeiro, C., and Thorup, M. A memetic algorithm for OSPF routing. In *Sixth INFORMS Telecommunications Conference, March 10-13, 2002 Hilton Deerfield Beach, Boca Raton, Florida* (2002), pp. 187–188.

[24] Burke, E., Jackson, K., Kingston, J., and Weare, R. Automated timetabling: The state of the art. *The Computer Journal 40*, 9 (1997), 565–571.

[25] Burke, E., and Newall, J. A phased evolutionary approach for the timetable problem: An initial study. In *Proceedings of the ICONIP/ANZIIS/ANNES '97 Conference* (Dunedin, New Zealand, 1997), Springer-Verlag, pp. 1038–1041.

[26] Burke, E., Newall, J., and Weare, R. A memetic algorithm for university exam timetabling. In *The Practice and Theory of Automated Timetabling*, E. Burke and P. Ross, Eds., vol. 1153 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996, pp. 241–250.

[27] Burke, E., Newall, J., and Weare, R. Initialisation strategies and diversity in evolutionary timetabling. *Evolutionary Computation 6*, 1 (1998), 81–103.

[28] Burke, E., and Smith, A. A memetic algorithm for the maintenance scheduling problem. In *Proceedings of the ICONIP/ANZIIS/ANNES '97 Conference* (Dunedin, New Zealand, 1997), Springer-Verlag, pp. 469–472.

[29] Burke, E., and Smith, A. A memetic algorithm to schedule grid maintenance. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation, Vi-*

enna: Evolutionary Computation and Fuzzy Logic for Intelligent Control, Knowledge Acquisition and Information Retrieval (1999), IOS Press 1999, pp. 122–127.

[30] Burke, E., and Smith, A. A multi-stage approach for the thermal generator maintenance scheduling problem. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Piscataway, NJ, USA, 1999), IEEE, pp. 1085–1092.

[31] Burke, E. K., Elliman, D. G., and Weare, R. F. A hybrid genetic algorithm for highly constrained timetabling problems. In *Proceedings of the Sixth International Conference on Genetic Algorithms* (1995), Morgan Kaufmann, San Francisco, CA, pp. 605–610.

[32] Cadieux, S., Tanizaki, N., and Okamura, T. Time efficient and robust 3-D brain image centering and realignment using hybrid genetic algorithm. In *Proceedings of the 36th SICE Annual Conference* (1997), IEEE, pp. 1279–1284.

[33] Caorsi, S., Massa, A., Pastorino, M., Rafetto, M., and Randazzo, A. A new approach to microwave imaging based on a memetic algorithm. In *PIERS 2002, Progress in Electromagnetics Research Symposium July 1-4, 2002, Cambridge, Massachusetts, USA* (2002). Invited.

[34] Carr, R., Hart, W., Krasnogor, N., Hirst, J., Burke, E., and Smith, J. Alignment of protein structures with a memetic evolutionary algorithm. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference* (New York, 9-13 July 2002), W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds., Morgan Kaufmann Publishers, pp. 1027–1034.

[35] Carrizo, J., Tinetti, F., and Moscato, P. A computational ecology for the quadratic assignment problem. In *Proceedings of the 21st Meeting on Informatics and Operations Research* (Buenos Aires, Argentina, 1992), SADIO.

[36] Cavalieri, S., and Gaiardelli, P. Hybrid genetic algorithms for a multiple-objective scheduling problem. *Journal of Intelligent Manufacturing 9*, 4 (1998), 361–367.

[37] Chaiyaratana, N., and Zalzala, A. Hybridisation of neural networks and genetic algorithms for time-optimal control. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington D.C., 1999), IEEE, pp. 389–396.

[38] Cheng, R., and Gen, M. Parallel machine scheduling problems using memetic algorithms. In *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems* (New York, NY, 1996), vol. 4, IEEE, pp. 2665–2670.

[39] Cheng, R., and Gen, M. Parallel machine scheduling problems using memetic algorithms. *Computers & Industrial Engineering 33*, 3–4 (1997), 761–764.

[40] Cheng, R., Gen, M., and Tsujimura, Y. A tutorial survey of job-shop scheduling problems using genetic algorithms. II. Hybrid genetic search strategies. *Computers & Industrial Engineering 37*, 1-2 (1999), 51–55.

[41] Chu, P., and Beasley, J. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research 24* (1997), 17–23.

[42] Ciuprina, G., Ioan, D., and Munteanu, I. Use of intelligent-particle swarm optimization in electromagnetics. *IEEE Transactions on Magnetics 38*, 2 (2002), 1037–1040.

[43] Clark, D., and Westhead, D. Evolutionary algorithms in computer-aided molecular design. *Journal of Computer-aided Molecular Design 10*, 4 (1996), 337–358.

[44] Coll, P., Durán, G., and Moscato, P. On worst-case and comparative analysis as design principles for efficient recombination operators: A graph coloring case study. In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999, pp. 279–294.

[45] Conradie, A., Mikkulainen, R., and Aldrich, C. Intelligent process control utilising symbiotic memetic neuro-evolution. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation, CEC'02, May 12-17, 2002, Honolulu, Hawaii, USA* (2002), X. Yao, Ed., pp. 623–628.

[46] Costa, D. An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR 33*, 3 (1995), 161–178.

[47] Costa, D., Dubuis, N., and Hertz, A. Embedding of a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *Journal of Heuristics 1*, 1 (1995), 105–128.

[48] Cotta, C. On resampling in nature-inspired heuristics. In *Proceedings of the 7^{th} Conference of the Spanish Association for Artificial Intelligence* (1997), V. Botti, Ed., pp. 145–154. In Spanish.

[49] Cotta, C. A study of hybridisation techniques and their application to the design of evolutionary algorithms. *AI Communications 11*, 3-4 (1998), 223–224.

[50] Cotta, C., and Moscato, P. Inferring phylogenetic trees using evolutionary algorithms. In *Parallel Problem Solving From Nature VII*, J. Merelo et al., Eds., vol. 2439 of *Lecture Notes in Computer Science*. Springer-Verlag, Paris, 2002.

[51] Cotta, C., and Muruzábal, J. Towards a more efficient evolutionary induction of bayesian networks. In *Parallel Problem Solving From Nature VII*, J. Merelo et al., Eds., vol. 2439 of *Lecture Notes in Computer Science*. Springer-Verlag, Paris, 2002.

[52] Cotta, C., and Troya, J. Genetic forma recombination in permutation flowshop problems. *Evolutionary Computation 6*, 1 (1998), 25–44.

[53] Cotta, C., and Troya, J. A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In *Artificial Neural Nets and Genetic Algorithms 3* (Wien New York, 1998), G. Smith, N. Steele, and R. Albrecht, Eds., Springer-Verlag, pp. 251–255.

[54] Cotta, C., and Troya, J. Using a hybrid evolutionary-A* approach for learning reactive behaviors. In *Real-World Applications of Evolutionary Computation* (Edinburgh, 15-16 Apr. 2000), S. Cagnoni et al., Eds., vol. 1803 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 347–356.

[55] Cotta, C., and Troya, J. A comparison of several evolutionary heuristics for the frequency assignment problem. In *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*, J. Mira and A. Prieto, Eds., vol. 2084 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, 2001, pp. 709–716.

[56] Cotta, C., and Troya, J. Embedding branch and bound within evolutionary algorithms. *Applied Intelligence* (2002). In press.

[57] Crain, T., Bishop, R., Fowler, W., and Rock, K. Optimal interplanetary trajectory design via hybrid genetic algorithm/recursive quadratic program search. In *Ninth AAS/AIAA Space Flight Mechanics Meeting* (Breckenridge CO, 1999), pp. 99–133.

[58] Dandekar, T., and Argos, P. Identifying the tertiary fold of small proteins with different topologies from sequence and secondary structure using the genetic algorithm and extended criteria specific for strand regions. *Journal of Molecular Biology 256*, 3 (1996), 645–660.

[59] Davidor, Y. Epistasis variance: A viewpoint on GA-hardness. In *Foundations of Genetic Algorithms*, G. Rawlins, Ed. Morgan Kaufmann, 1991, pp. 23–35.

[60] Davidor, Y., and Ben-Kiki, O. The interplay among the genetic algorithm operators: Information theory tools used in a holistic way. In *Parallel Problem Solving From Nature II* (Amsterdam, 1992), R. Männer and B. Manderick, Eds., Elsevier Science Publishers B.V., pp. 75–84.

[61] Davis, L. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold Computer Library, New York, 1991.

[62] Dawkins, R. *The Selfish Gene*. Clarendon Press, Oxford, 1976.

[63] de Causmaecker, P., van den Berghe, G., and Burke, E. Using tabu search as a local heuristic in a memetic algorithm for the nurse rostering problem. In *Proceedings of the Thirteenth Conference on Quantitative Methods for Decision Making* (Brussels, Belgium, 1999), pp. abstract only, poster presentation.

[64] de Souza, P., Garg, R., and Garg, V. Automation of the analysis of Mossbauer spectra. *Hyperfine Interactions 112*, 1–4 (1998), 275–278.

[65] Deaven, D., and Ho, K. Molecular-geometry optimization with a genetic algorithm. *Physical Review Letters 75*, 2 (1995), 288–291.

[66] Deaven, D., Tit, N., Morris, J., and Ho, K. Structural optimization of Lennard-Jones clusters by a genetic algorithm. *Chemical Physics Letters 256*, 1–2 (1996), 195–200.

[67] Dellaert, N., and Jeunet, J. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research 38*, 5 (2000), 1083–1099.

[68] Desjarlais, J., and Handel, T. New strategies in protein design. *Current Opinion in Biotechnology 6*, 4 (1995), 460–466.

[69] Doll, R., and VanHove, M. Global optimization in LEED structure determination using genetic algorithms. *Surface Science 355*, 1-3 (1996), L393–L398.

[70] Dorne, R., and Hao, J. A new genetic local search algorithm for graph coloring. In *Parallel Problem Solving From Nature V* (Berlin, 1998), A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 745–754.

[71] dos Santos Coelho, L., Rudek, M., and Junior, O. C. Fuzzy-memetic approach for prediction of chaotic time series and nonlinear identification. In *6th On-line World Conference on Soft Computing in Industrial Applications, Organized by World Federation of Soft Computing* (2001). Co-sponred by IEEE Systems, Man, and Cybenertics Society.

[72] Eiben, A., Raue, P.-E., and Ruttkay, Z. Genetic algorithms with multi-parent recombination. In *Parallel Problem Solving From Nature III*, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994, pp. 78–87.

[73] Fang, J., and Xi, Y. A rolling horizon job shop rescheduling strategy in the dynamic environment. *International Journal of Advanced Manufacturing Technology 13*, 3 (1997), 227–232.

[74] Fleurent, C., and Ferland, J. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research 63* (1997), 437–461.

[75] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, 1966.

[76] França, P., Mendes, A., and Moscato, P. Memetic algorithms to minimize tardiness on a single machine with sequence-dependent setup times. In *Proceedings of the 5th International Conference of the Decision Sciences Institute, Athens, Greece* (Atlanta, GA, USA, 1999), Decision Sciences Institute, pp. 1708–1710.

[77] França, P., Mendes, A., and Moscato, P. A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research 132*, 1 (2001), 224–242.

[78] Freisleben, B., and Merz, P. A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation* (Piscataway, NJ, USA, 1996), T. Bäck, H. Kitano, and Z. Michalewicz, Eds., IEEE Press, pp. 616–621.

[79] Freisleben, B., and Merz, P. New Genetic Local Search Operators for the Traveling Salesman Problem. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature - PPSN IV* (Berlin, Germany, 1996), H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., vol. 1141 of *Lecture Notes in Computer Science*, Springer, pp. 890–900.

[80] Fu, R., Esfarjani, K., Hashi, Y., Wu, J., Sun, X., and Kawazoe, Y. Surface reconstruction of Si (001) by genetic algorithm and simulated annealing method. *Science Reports of The Research Institutes Tohoku University Series A-Physics Chemistry And Metallurgy 44*, 1 (Mar. 1997), 77–81.

[81] Garcia, B., Mahey, P., and LeBlanc, L. Iterative improvement methods for a multiperiod network design problem. *European Journal of Operational Research 110*, 1 (1998), 150–165.

[82] Gen, M., Ida, K., and Yinzhen, L. Bicriteria transportation problem by hybrid genetic algorithm. *Computers & Industrial Engineering 35*, 1-2 (1998), 363–366.

[83] Glover, F., and Laguna, M. *Tabu Search*. Kluwer Academic Publishers, Boston, MA, 1997.

[84] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[85] Goldberg, D., and Lingle Jr., R. Alleles, loci and the traveling salesman problem. In *Proceedings of an International Conference on Genetic Algorithms* (Hillsdale NJ, 1985), J. Grefenstette, Ed., Lawrence Erlbaum Associates.

[86] Goldstein, A., and Lesk, A. Common feature techniques for discrete optimization. Comp. Sci. Tech. Report 27, Bell. Tel. Labs, March 1975.

[87] Gonçalves, J. F. A memetic algorithm for the examination timetabling problem. In *Optimization 2001, Aveiro, Portugal, July 23-25, 2001* (2001).

[88] Gorges-Schleuter, M. ASPARAGOS: An asynchronous parallel genetic optimization strategy. In *Proceedings of the Third International Conference on Genetic Algorithms* (1989), J. D. Schaffer, Ed., Morgan Kaufmann Publishers, pp. 422–427.

[89] Gorges-Schleuter, M. *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. PhD thesis, University of Dortmund, Germany, 1991.

[90] Gorges-Schleuter, M. Asparagos96 and the traveling salesman problem. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, Indianapolis, USA* (Piscataway, NJ, USA, 1997), T. Bäck, Z. Michalewicz, and X. Yao, Eds., IEEE Press, pp. 171–174.

[91] Gottlieb, J. Permutation-based evolutionary algorithms for multidimensional knapsack problems. In *ACM Symposium on Applied Computing 2000* (2000), J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, Eds., ACM Press, pp. 408–414.

[92] Grimbleby, J. Hybrid genetic algorithms for analogue network synthesis. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington D.C., 1999), IEEE, pp. 1781–1787.

[93] Gunn, J. Sampling protein conformations using segment libraries and a genetic algorithm. *Journal of Chemical Physics 106*, 10 (1997), 4270–4281.

[94] Guotian, M., and Changhong, L. Optimal design of the broadband stepped impedance transformer based on the hybrid genetic algorithm. *Journal of Xidian University 26*, 1 (1999), 8–12.

[95] Haas, O., Burnham, K., and Mills, J. Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology 43*, 8 (1998), 2179–2193.

[96] Haas, O., Burnham, K., Mills, J., Reeves, C., and Fisher, M. Hybrid genetic algorithms applied to beam orientation in radiotherapy. In *Fourth European Congress on Intelligent Techniques and Soft Computing Proceedings* (Aachen, Germany, 1996), vol. 3, Verlag Mainz, pp. 2050–2055.

[97] Hadj-Alouane, A., Bean, J., and Murty, K. A hybrid genetic/optimization algorithm for a task allocation problem. *Journal of Scheduling 2*, 4 (1999).

[98] Hansen, P., and Mladenović, N. Variable neighborhood search: Principles and applications. *European Journal of Operational Research 130*, 3 (2001), 449–467.

[99] Harris, S., and Ifeachor, E. Automatic design of frequency sampling filters by hybrid genetic algorithm techniques. *IEEE Transactions on Signal Processing 46*, 12 (1998), 3304–3314.

[100] Hart, W., and Belew, R. Optimizing an arbitrary function is hard for the genetic algorithm. In *Proceedings of the 4^{th} International Conference on Genetic Algorithms* (San Mateo CA, 1991), R. Belew and L. Booker, Eds., Morgan Kaufmann, pp. 190–195.

[101] Hartke, B. Global geometry optimization of clusters using genetic algorithms. *Journal of Physical Chemistry 97*, 39 (1993), 9973–9976.

[102] Hifi, M. A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems. *Journal of the Operational Research Society 48*, 6 (1997), 612–622.

[103] Hirsch, R., and Mullergoymann, C. Fitting of diffusion-coefficients in a 3-compartment sustained-release drug formulation using a genetic algorithm. *International Journal of Pharmaceutics 120*, 2 (1995), 229–234.

[104] Ho, K., Shvartsburg, A., Pan, B., Lu, Z., Wang, C., Wacker, J., Fye, J., and Jarrold, M. Structures of medium-sized silicon clusters. *Nature 392*, 6676 (1998), 582–585.

[105] Hobday, S., and Smith, R. Optimisation of carbon cluster geometry using a genetic algorithm. *Journal of The Chemical Society-Faraday Transactions 93*, 22 (1997), 3919–3926.

[106] Hodgson, R. Memetic algorithms and the molecular geometry optimization problem. In *Proceedings of the 2000 Congress on Evolutionary Computation* (Piscataway, NJ, 2000), IEEE Service Center, pp. 625–632.

[107] Hodgson, R. Memetic algorithm approach to thin-film optical coating design. In *Second Workshop on Memetic Algorithms (2nd WOMA)* (San Francisco, California, USA, 7 July 2001), W. Hart, N. Krasnogor, and J. Smith, Eds., pp. 152–157.

[108] Holland, J. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[109] Holstein, D., and Moscato, P. Memetic algorithms using guided local search: A case study. In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999, pp. 235–244.

[110] Hopper, E., and Turton, B. A genetic algorithm for a 2d industrial packing problem. *Computers & Industrial Engineering 37*, 1-2 (1999), 375–378.

[111] Hulin, M. An optimal stop criterion for genetic algorithms: A bayesian approach. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Mateo, CA, 1997), T. Bäck, Ed., Morgan Kaufmann, pp. 135–143.

[112] Ichimura, T., and Kuriyama, Y. Learning of neural networks with parallel hybrid GA using a Royal Road function. In *1998 IEEE International Joint Conference on Neural Networks* (New York, NY, 1998), vol. 2, IEEE, pp. 1131–1136.

[113] Jih, W., and Hsu, Y. Dynamic vehicle routing using hybrid genetic algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington D.C., 1999), IEEE, pp. 453–458.

[114] Jones, G., Willett, P., Glen, R., Leach, A., and Taylor, R. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology 267*, 3 (1997), 727–748.

[115] Jones, T. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, 1995.

[116] Kariuki, B., Serrano-Gonzalez, H., Johnston, R., and Harris, K. The application of a genetic algorithm for solving crystal structures from powder diffraction data. *Chemical Physics Letters 280*, 3-4 (1997), 189–195.

[117] Karp, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum, New York NY, 1972, pp. 85–103.

[118] Kassotakis, I., Markaki, M., and Vasilakos, A. A hybrid genetic approach for channel reuse in multiple access telecommunication networks. *IEEE Journal on Selected Areas in Communications 18*, 2 (2000), 234–243.

[119] Katayama, K., Hirabayashi, H., and Narihisa, H. Performance analysis for crossover operators of genetic algorithm. *Transactions of the Institute of Electronics, Information and Communication Engineers J81D-I*, 6 (1998), 639–650.

[120] Kersting, S., Raidl, G., and Ljubić, I. A memetic algorithm for vertex-biconnectivity augmentation. In *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim* (Kinsale, Ireland, 3-4 April 2002), S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, Eds., vol. 2279 of *LNCS*, Springer-Verlag, pp. 101–110.

[121] Kim, T., and May, G. Intelligent control of via formation by photosensitive BCB for MCM-L/D applications. *IEEE Transactions on Semiconductor Manufacturing 12* (1999), 503–515.

[122] Kirkpatrick, S., Gelatt Jr., C., and Vecchi, M. Optimization by simulated annealing. *Science 220*, 4598 (1983), 671–680.

[123] Knödler, K., Poland, J., Zell, A., and Mitterer, A. Memetic algorithms for combinatorial optimization problems in the calibration of modern combustion engines. In *GECCO 2002: Proceedings*

of the Genetic and Evolutionary Computation Conference (New York, 9-13 July 2002), W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds., Morgan Kaufmann Publishers, p. 687.

[124] KRASNOGOR, N. Coevolution of genes and memes in memetic algorithms. In *Graduate Student Workshop* (Orlando, Florida, USA, 13 July 1999), U.-M. O'Reilly, Ed., p. 371.

[125] KRASNOGOR, N. *Studies on the Theory and Design Space of Memetic Algorithms*. Ph.D. Thesis, Faculty of Engineering, Computer Science and Mathematics. University of the West of England. Bristol, United Kingdom, 2002.

[126] KRASNOGOR, N., BLACKBURNE, B., BURKE, E. K., AND HIRST, J. D. Multimeme algorithms for protein structure prediction. In *Proceedings of the Parallel Problem Solving from Nature VII. Lecture notes in computer science* (2002). to appear.

[127] KRASNOGOR, N., BLACKBURNE, B., HIRST, J., AND BURKE, E. Multimeme algorithms for protein structure prediction. In *7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain* (2002).

[128] KRASNOGOR, N., AND SMITH, J. A memetic algorithm with self-adaptive local search: TSP as a case study. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)* (Las Vegas, Nevada, USA, 10-12 July 2000), D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds., Morgan Kaufmann, pp. 987–994.

[129] KRASNOGOR, N., AND SMITH, J. Emergence of profitable search strategies based on a simple inheritance mechanism. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (San Francisco, California, USA, 7-11 July 2001), L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, Eds., Morgan Kaufmann, pp. 432–439.

[130] KRASNOGOR, N., AND SMITH, J. Emergence of profitable search strategies based on a simple inheritance mechanism. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (San Francisco, California, USA, 7-11 July 2001), L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds., Morgan Kaufmann, pp. 432–439.

[131] Krasnogor, N., and Smith, J. Multimeme algorithms for the structure prediction and structure comparison of proteins. In *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference* (New York, 8 July 2002), A. M. Barry, Ed., AAAI, pp. 42–44.

[132] Krishna, K., and Narasimha-Murty, M. Genetic k-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 29*, 3 (1999), 433–439.

[133] Krishna, K., Ramakrishnan, K., and Thathachar, M. Vector quantization using genetic k-means algorithm for image compression. In *1997 International Conference on Information, Communications and Signal Processing* (New York, NY, 1997), vol. 3, IEEE, pp. 1585–1587.

[134] Krzanowski, R., and Raper, J. Hybrid genetic algorithm for transmitter location in wireless networks. *Computers, Environment and Urban Systems 23*, 5 (1999), 359–382.

[135] Landree, E., Collazo-Davila, C., and Marks, L. Multi-solution genetic algorithm approach to surface structure determination using direct methods. *Acta Crystallographica Section B - Structural Science 53* (1997), 916–922.

[136] Lazar, G., Desjarlais, J., and Handel, T. De novo design of the hydrophobic core of ubiquitin. *Protein Science 6*, 6 (1997), 1167–1178.

[137] Lee, C. Genetic algorithms for single machine job scheduling with common due date and symmetric penalties. *Journal of the Operations Research Society of Japan 37*, 2 (1994), 83–95.

[138] Levine, D. A parallel genetic algorithm for the set partitioning problem. In *Meta-Heuristics: Theory & Applications*, I. Osman and J. Kelly, Eds. Kluwer Academic Publishers, Boston, MA, USA, 1996, pp. 23–35.

[139] Li, F., Morgan, R., and Williams, D. Economic environmental dispatch made easy with hybrid genetic algorithms. In *Proceedings of the International Conference on Electrical Engineering* (Beijing, China, 1996), vol. 2, Int. Acad. Publishers, pp. 965–969.

[140] Li, L., Darden, T., Freedman, S., Furie, B., Furie, B., Baleja, J., Smith, H., Hiskey, R., and Pedersen, L. Refinement of the NMR solution structure of the gamma-carboxyglutamic acid domain of coagulation factor IX using molecular dynamics simulation with initial Ca2+ positions determined by a genetic algorithm. *Biochemistry 36*, 8 (1997), 2132–2138.

[141] Li, S. Toward global solution to map image estimation: Using common structure of local solutions. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 1223 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, 1997, pp. 361–374.

[142] Liaw, C. A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Oprational Research 124*, 1 (2000), 28–42.

[143] Lin, S. Computer solutions of the traveling salesman problem. *Bell System Technical Journal 10* (December 1965), 2245–2269.

[144] Lin, S., and Kernighan, B. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research 21* (1973), 498–516.

[145] Ling, S. E. Integrating genetic algorithms with a prolog assignment program as a hybrid solution for a polytechnic timetable problem. In *Parallel Problem Solving from Nature II.* Elsevier Science Publisher B. V., 1992, pp. 321–329.

[146] Lorber, D., and Shoichet, B. Flexible ligand docking using conformational ensembles. *Protein Science 7*, 4 (1998), 938–950.

[147] Louis, S., Yin, X., and Yuan, Z. Multiple vehicle routing with time windows using genetic algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington D.C., 1999), IEEE Neural Network Council - Evolutionary Programming Society - Institution of Electrical Engineers, pp. 1804–1808.

[148] MacKay, A. Generalized crystallography. *THEOCHEM-Journal of Molecular Structure 336*, 2–3 (1995), 293–303.

[149] Maddox, J. Genetics helping molecular-dynamics. *Nature 376*, 6537 (1995), 209–209.

[150] Mathias, K., and Whitley, D. Genetic operators, the fitness landscape and the traveling salesman problem. In *Parallel Problem Solving From Nature II* (Amsterdam, 1992), R. Männer and B. Manderick, Eds., Elsevier Science Publishers B.V., p. **buscar!!!**

[151] Mathias, K., and Whitley, L. Noisy function evaluation and the delta coding algorithm. In *Proceedings of the SPIE–The International Society for Optical Engineering* (1994), pp. 53–64.

[152] May, A., and Johnson, M. Protein-structure comparisons using a combination of a genetic algorithm, dynamic-programming and least-squares minimization. *Protein Engineering 7*, 4 (1994), 475–485.

[153] Mendes, A., França, P., and Moscato, P. Fitness landscapes for the total tardiness single machine scheduling problem. *Neural Network World, an International Journal on Neural and Mass-Parallel Computing and Information Systems* (2002), 165–180.

[154] Mendes, A., Muller, F., França, P., and Moscato, P. Comparing meta-heuristic approaches for parallel machine scheduling problems with sequence-dependent setup times. In *Proceedings of the 15th International Conference on CAD/CAM Robotics & Factories of the Future, Aguas de Lindoia, Brasil* (Campinas, SP, Brazil, 1999), vol. 1, Technological Center for Informatics Foundation, pp. 1–6.

[155] Merkle, L., Lamont, G., Gates, G. J., and Pachter, R. Hybrid genetic algorithms for minimization of a polypeptide specific energy model. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation* (New York, NY, 1996), IEEE, pp. 396–400.

[156] Merz, P. A comparison of memetic recombination operators for the traveling salesman problem. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference* (New York, 9-13 July 2002), W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, Eds., Morgan Kaufmann Publishers, pp. 472–479.

[157] Merz, P., and Freisleben, B. A Genetic Local Search Approach to the Quadratic Assignment Problem. In *Proceedings of the 7th International Conference on Genetic Algorithms* (San Francisco, CA, USA, 1997), T. Bäck, Ed., Morgan Kaufmann, pp. 465–472.

[158] Merz, P., and Freisleben, B. Genetic Local Search for the TSP: New Results. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation* (Piscataway, NJ, USA, 1997), T. Bäck, Z. Michalewicz, and X. Yao, Eds., IEEE Press, pp. 159–164.

[159] Merz, P., and Freisleben, B. Memetic Algorithms and the Fitness Landscape of the Graph Bi-Partitioning Problem. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN V* (Berlin, Germany, 1998), A.-E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498 of *Lecture Notes in Computer Science*, Springer, pp. 765–774.

[160] Merz, P., and Freisleben, B. On the Effectiveness of Evolutionary Search in High–Dimensional NK-Landscapes. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation* (Piscataway, NJ, USA, 1998), D. Fogel, Ed., IEEE Press, pp. 741–745.

[161] Merz, P., and Freisleben, B. A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem. In *1999 Congress on Evolutionary Computation (CEC'99)* (Piscataway, NJ, USA, 1999), P. Angeline, Ed., IEEE Press, pp. 2063–2070.

[162] Merz, P., and Freisleben, B. Fitness landscapes and memetic algorithm design. In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999, pp. 245–260.

[163] Merz, P., and Freisleben, B. Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. *Evolutionary Computation 8*, 1 (2000), 61–91.

[164] Merz, P., and Freisleben, B. Greedy and local search heuristics for the unconstrained binary quadratic programming problem. *Journal of Heuristics 8*, 2 (2002), 197–213.

[165] Merz, P., and Freisleben, B. Memetic algorithms for the traveling salesman problem. *Complex Systems* (2002). to be published.

[166] Merz, P., and Katayama, K. Memetic algorithms for the unconstrained binary quadratic programming problem. *Bio Systems* (2002). to be published.

[167] Merz, P., and Zell, A. Clustering gene expression profiles with memetic algorithms. In *7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain* (2002).

[168] Meza, J., Judson, R., Faulkner, T., and Treasurywala, A. A comparison of a direct search method and a genetic algorithm for conformational searching. *Journal of Computational Chemistry 17*, 9 (1996), 1142–1151.

[169] Mignotte, M., Collet, C., Pérez, P., and Bouthemy, P. Hybrid genetic optimization and statistical model based approach for the classification of shadow shapes in sonar imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 2 (2000), 129–141.

[170] Miller, D., Chen, H., Matson, J., and Liu, Q. A hybrid genetic algorithm for the single machine scheduling problem. *Journal of Heuristics 5*, 4 (1999), 437–454.

[171] Miller, S., Hogle, J., and Filman, D. A genetic algorithm for the *ab initio* phasing of icosahedral viruses. *Acta Crystallographica Section D - Biological Crystallography 52* (1996), 235–251.

[172] Min, L., and Cheng, W. Identical parallel machine scheduling problem for minimizing the makespan using genetic algorithm combined with simulated annealing. *Chinese Journal of Electronics 7*, 4 (1998), 317–321.

[173] Ming, X., and Mak, K. A hybrid hopfield network-genetic algorithm approach to optimal process plan selection. *International Journal of Production Research 38*, 8 (2000), 1823–1839.

[174] Monfroglio, A. Hybrid genetic algorithms for a rostering problem. *Software – Practice and Experience 26*, 7 (1996), 851–862.

[175] Monfroglio, A. Hybrid genetic algorithms for timetabling. *International Journal of Intelligent Systems 11*, 8 (1996), 477–523.

[176] Monfroglio, A. Timetabling through constrained heuristic search and genetic algorithms. *Software – Practice and Experience 26*, 3 (1996), 251–279.

[177] Moscato, P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Tech. Rep. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.

[178] Moscato, P. On genetic crossover operators for relative order preservation. C3P Report 778, California Institute of Technology, Pasadena, CA 91125, 1989.

[179] Moscato, P. An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: The Role of Tabu Search. *Annals of Operations Research 41*, 1-4 (1993), 85–121.

[180] Moscato, P. Memetic algorithms: A short introduction. In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999, pp. 219–234.

[181] Moscato, P., and Norman, M. G. A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems. In *Parallel Computing and Transputer Applications* (Amsterdam, 1992), M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, Eds., IOS Press, pp. 177–186.

[182] Moscato, P., and Tinetti, F. Blending heuristics with a population-based approach: A memetic algorithm for the traveling salesman problem. Report 92-12, Universidad Nacional de La Plata, C.C. 75, 1900 La Plata, Argentina, 1992.

[183] Murata, T., and Ishibuchi, H. Performance evaluation of genetic algorithms for flowshop scheduling problems. In *Proceedings of the First IEEE Conference on Evolutionary Computation* (New York, NY, 1994), vol. 2, IEEE, pp. 812–817.

[184] Murata, T., Ishibuchi, H., and Tanaka, H. Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering 30*, 4 (1996), 1061–1071.

[185] Musil, M., Wilmut, M., and Chapman, N. A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion. *IEEE Journal of Oceanic Engineering 24*, 3 (1999), 358–369.

[186] Nagata, Y., and Kobayashi, S. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In *Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, EUA* (San Mateo, CA, 1997), T. Bäck, Ed., Morgan Kaufmann, pp. 450–457.

[187] Niesse, J., and Mayne, H. Global geometry optimization of atomic clusters using a modified genetic algorithm in space-fixed coordinates. *Journal of Chemical Physics 105*, 11 (1996), 4700–4706.

[188] Nordstrom, A., and Tufekci, S. A genetic algorithm for the talent scheduling problem. *Computers & Operations-Research 21*, 8 (1994), 927–940.

[189] Norman, M., and Moscato, P. A competitive and cooperative approach to complex combinatorial search. Tech. Rep. Caltech Concurrent Computation Program, Report. 790, California Institute of Technology, Pasadena, California, USA, 1989. expanded version published at the Proceedings of the 20th Informatics and Operations Research Meeting, Buenos Aires (20th JAIIO), Aug. 1991, pp. 3.15–3.29.

[190] Novaes, A., De-Cursi, J., and Graciolli, O. A continuous approach to the design of physical distribution systems. *Computers & Operations Research 27*, 9 (2000), 877–893.

[191] Oliver, I., Smith, D., and Holland, J. A study of permutation crossover operators on the traveling salesperson problem. In *Proceedings of the 2^{nd} International Conference on Genetic Algorithms and their Applications* (Hillsdale NJ, 1987), J. Grefenstette, Ed., Lawrence Erlbaum Associates, pp. 224–230.

[192] Osmera, P. Hybrid and distributed genetic algorithms for motion control. In *Proceedings of the Fourth International Symposium*

on Measurement and Control in Robotics (1995), V. Chundy and E. Kurekova, Eds., pp. 297–300.

[193] Ostermark, R. A neuro-genetic algorithm for heteroskedastic time-series processes: empirical tests on global asset returns. *Soft Computing 3*, 4 (1999), 206–220.

[194] Ostermark, R. Solving a nonlinear non-convex trim loss problem with a genetic hybrid algorithm. *Computers & Operations Research 26*, 6 (1999), 623–635.

[195] Ostermark, R. Solving irregular econometric and mathematical optimization problems with a genetic hybrid algorithm. *Computational Economics 13*, 2 (1999), 103–115.

[196] Ozcan, E., and Mohan, C. Steady state memetic algorithm for partial shape matching. In *Evolutionary Programming VII* (1998), V. Porto, N. Saravanan, and D. Waagen, Eds., vol. 1447 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 527–536.

[197] Ozdamar, L. A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 29*, 1 (1999), 44–59.

[198] Pacey, M., Patterson, E., and James, M. A photoelastic technique for characterising fatigue crack closure and the effective stress intensity factor. *Zeszyty Naukowe Politechniki Opolskiej, Seria: Mechanika z.67*, Nr kol. 269/2001 (2001). *VII Summer School of Fracture Mechanic, Current Research in Fatigue and Fracture*, Pokrzywna (Poland), 18-22 Jun. 2001.

[199] Pacey, M., Wang, X., Haake, S., and Patterson, E. The application of evolutionary and maximum entropy algorithms to photoelastic spectral analysis. *Experimental Mechanics 39*, 4 (1999), 265–273.

[200] Paechter, B., Cumming, A., Norman, M., and Luchian, H. Extensions to a Memetic timetabling system. In *The Practice and Theory of Automated Timetabling*, E. Burke and P. Ross, Eds., vol. 1153 of *Lecture Notes in Computer Science*. Springer Verlag, 1996, pp. 251–265.

[201] Paechter, B., Rankin, R., and Cumming, A. Improving a lecture timetabling system for university wide use. In *The Practice and Theory of Automated Timetabling II*, E. Burke and M. Carter, Eds., vol. 1408 of *Lecture Notes in Computer Science*. Springer Verlag, 1998, pp. 156–165.

[202] Papadimitriou, C., and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[203] Pastorino, M., Caorsi, S., Massa, A., and Randazzo, A. Reconstruction algorithms for electromagnetic imaging. In *Proceedings of IEEE Instrumentation and Measurement Technology Conf. (IMTC/2002), Anchorage, Alaska, USA, May, 2002* (2002), pp. 1695–1700. special session on Imaging Systems.

[204] Poland, J., Knödler, K., Mitterer, A., Fleischhauer, T., Zuber-Goos, F., and Zell, A. Evolutionary search for smooth maps in motor control unit calibration. In *Stochastic Algorithms: Foundations and Applications*, Steinhöfel, Ed., vol. 2264 of *LNCS*. Springer-Verlag, 2001, pp. 107–116.

[205] Pratihar, D., Deb, K., and Ghosh, A. Fuzzy-genetic algorithms and mobile robot navigation among static obstacles. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington D.C., 1999), IEEE, pp. 327–334.

[206] Pucello, N., Rosati, M., D'Agostino, G., Pisacane, F., Rosato, V., and Celino, M. Search of molecular ground state via genetic algorithm: Implementation on a hybrid SIMD-MIMD platform. *International Journal of Modern Physics C 8*, 2 (1997), 239–252.

[207] Pullan, W. Structure prediction of benzene clusters using a genetic algorithm. *Journal of Chemical Information and Computer Sciences 37*, 6 (1997), 1189–1193.

[208] Quagliarella, D., and Vicini, A. Hybrid genetic algorithms as tools for complex optimisation problems. In *New Trends in Fuzzy Logic II. Proceedings of the Second Italian Workshop on Fuzzy Logic* (Singapore, 1998), P. Blonda, M. Castellano, and A. Petrosino, Eds., World Scientific, pp. 300–307.

[209] Quintero, A., and Pierre, S. A multi-population memetic algorithm to optimize the assignment of cells to switches in cellular mobile networks, 2001. submitted for publication.

[210] Radcliffe, N. Non-linear genetic representations. In *Parallel Problem Solving From Nature 2*, R. Männer and B. Manderick, Eds. Elsevier Science Publishers, Amsterdam, 1992, pp. 259–268.

[211] Radcliffe, N. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence 10* (1994), 339–384.

[212] Radcliffe, N., and Surry, P. Fitness Variance of Formae and Performance Prediction. In *Proceedings of the 3^{rd} Workshop on Foundations of Genetic Algorithms* (San Francisco, 1994), L. Whitley and M. Vose, Eds., Morgan Kaufmann, pp. 51–72.

[213] Radcliffe, N., and Surry, P. Formal Memetic Algorithms. In *Evolutionary Computing: AISB Workshop* (1994), T. Fogarty, Ed., vol. 865 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 1–16.

[214] Raidl, G., and Julstron, B. A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In *ACM Symposium on Applied Computing 2000* (2000), J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, Eds., ACM Press, pp. 440–445.

[215] Ramat, E., Venturini, G., Lente, C., and Slimane, M. Solving the multiple resource constrained project scheduling problem with a hybrid genetic algorithm. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Francisco CA, 1997), T. Bäck, Ed., Morgan Kaufmann, pp. 489–496.

[216] Rankin, R. Automatic timetabling in practice. In *Practice and Theory of Automated Timetabling. First International Conference. Selected Papers* (Berlin, 1996), Springer-Verlag, pp. 266–279.

[217] Raymer, M., Sanschagrin, P., Punch, W., Venkataraman, S., Goodman, E., and Kuhn, L. Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm. *Journal of Molecular Biology 265*, 4 (1997), 445–464.

[218] Rechenberg, I. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann-Holzboog, Stuttgart, 1973.

[219] Reeves, C. Hybrid genetic algorithms for bin-packing and related problems. *Annals of Operations Research 63* (1996), 371–396.

[220] Reich, C. Simulation of imprecise ordinary differential equations using evolutionary algorithms. In *ACM Symposium on Applied Computing 2000* (2000), J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, Eds., ACM Press, pp. 428–432.

[221] Ridao, M., Riquelme, J., Camacho, E., and Toro, M. An evolutionary and local search algorithm for planning two manipulators motion. In *Tasks and Methods in Applied Artificial Intelligence*,

A. Del Pobil, J. Mira, and M. Ali, Eds., vol. 1416 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, 1998, pp. 105–114.

[222] Rodrigues, A., and Ferreira, J. S. Solving the rural postman problem by memetic algorithms. In *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001* (2001), J. P. de Sousa, Ed., pp. 679–684.

[223] Ruff, C., Hughes, S., and Hawkes, D. Volume estimation from sparse planar images using deformable models. *Image and Vision Computing 17*, 8 (1999), 559–565.

[224] Runggeratigul, S. A memetic algorithm to communication network design taking into consideration an existing network. In *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001* (2001), J. P. de Sousa, Ed., pp. 91–96.

[225] Sakamoto, A., Liu, X., and Shimamoto, T. A genetic approach for maximum independent set problems. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E80A*, 3 (1997), 551–556.

[226] Schnecke, V., and Vornberger, O. Hybrid genetic algorithms for constrained placement problems. *IEEE Transactions on Evolutionary Computation 1*, 4 (1997), 266–277.

[227] Schwefel, H.-P. *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik*. Diplomarbeit, Technische Universität Berlin, Hermann Föttinger–Institut für Strömungstechnik, März 1965.

[228] Shankland, K., David, W., and Csoka, T. Crystal structure determination from powder diffraction data by the application of a genetic algorithm. *Zeitschrift Fur Kristallographie 212*, 8 (1997), 550–552.

[229] Shankland, K., David, W., Csoka, T., and McBride, L. Structure solution of ibuprofen from powder diffraction data by the application of a genetic algorithm combined with prior conformational analysis. *International Journal of Pharmaceutics 165*, 1 (1998), 117–126.

[230] Smith, J. Co-evolving memetic algorithms: Initial investigations. In *7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain* (2002).

[231] Srinivasan, D., Cheu, R., Poh, Y., and Ng, A. Development of an intelligent technique for traffic network incident detection. *Engineering Applications of Artificial Intelligence 13*, 3 (2000), 311–322.

[232] Surry, P., and Radcliffe, N. Inoculation to initialise evolutionary search. In *Evolutionary Computing: AISB Workshop*, T. Fogarty, Ed., no. 1143 in Lecture Notes in Computer Science. Springer-Verlag, 1996, pp. 269–285.

[233] Syswerda, G. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms* (San Mateo, CA, 1989), J. Schaffer, Ed., Morgan Kaufmann, pp. 2–9.

[234] Taguchi, T., Yokota, T., and Gen, M. Reliability optimal design problem with interval coefficients using hybrid genetic algorithms. *Computers & Industrial Engineering 35*, 1–2 (1998), 373–376.

[235] Tam, K., and Compton, R. GAMATCH - a genetic algorithm-based program for indexing crystal faces. *Journal of Applied Crystallography 28* (1995), 640–645.

[236] Topchy, A., Lebedko, O., and Miagkikh, V. Fast learning in multilayered networks by means of hybrid evolutionary and gradient algorithms. In *Proceedings of International Conference on Evolutionary Computation and its Applications* (June 1996), pp. 390–398.

[237] Urdaneta, A., Gómez, J., Sorrentino, E., Flores, L., and Díaz, R. A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming. *IEEE Transactions on Power Systems 14*, 4 (1999), 1292–1298.

[238] vanKampen, A., Strom, C., and Buydens, L. Lethalization, penalty and repair functions for constraint handling in the genetic algorithm methodology. *Chemometrics And Intelligent Laboratory Systems 34*, 1 (1996), 55–68.

[239] Wang, L., and Yen, J. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter. *Fuzzy Sets and Systems 101*, 3 (1999), 353–362.

[240] Watson, J., Rana, S., Whitley, L., and Howe, A. The impact of approximate evaluation on the performance of search algorithms for warehouse scheduling. *Journal of Scheduling 2*, 2 (1999), 79–98.

[241] Wehrens, R., Lucasius, C., Buydens, L., and Kateman, G. HIPS, A hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms. *Analytica Chimica ACTA 277*, 2 (May 1993), 313–324.

[242] Wei, P., and Cheng, L. A hybrid genetic algorithm for function optimization. *Journal of Software 10*, 8 (1999), 819–823.

[243] Wei, X., and Kangling, F. A hybrid genetic algorithm for global solution of nondifferentiable nonlinear function. *Control Theory & Applications 17*, 2 (2000), 180–183.

[244] Weile, D., and Michielssen, E. Design of doubly periodic filter and polarizer structures using a hybridized genetic algorithm. *Radio Science 34*, 1 (1999), 51–63.

[245] White, R., Niesse, J., and Mayne, H. A study of genetic algorithm approaches to global geometry optimization of aromatic hydrocarbon microclusters. *Journal of Chemical Physics 108*, 5 (1998), 2208–2218.

[246] Willett, P. Genetic algorithms in molecular recognition and design. *Trends in Biotechnology 13*, 12 (1995), 516–521.

[247] Wolpert, D., and Macready, W. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation 1(1)* (1997), 67–82.

[248] Xiao, J., and Zhang, L. Adaptive evolutionary planner/navigator for mobile robots. *IEEE Transactions on Evolutionary Computation 1*, 1 (1997), 18–28.

[249] Yao, X. Evolutionary artificial neural networks. *Int. Journal of Neural Systems 4*, 3 (Sept. 1993), 203–222.

[250] Yeh, I. Hybrid genetic algorithms for optimization of truss structures. *Computer Aided Civil and Infrastructure Engineering 14*, 3 (1999), 199–206.

[251] Yeh, W.-C. A memetic algorithm for the min k-cut problem. *Control and Intelligent Systems 28*, 2 (2000), 47–55.

[252] Yoneyama, M., Komori, H., and Nakamura, S. Estimation of impulse response of vocal tract using hybrid genetic algorithm-a case of only glottal source. *Journal of the Acoustical Society of Japan 55*, 12 (1999), 821–830.

[253] Zacharias, C., Lemes, M., and Pino, A. Combining genetic algorithm and simulated annealing: a molecular geometry optimization study. *THEOCHEM-Journal of Molecular Structure 430*, 29–39 (1998).

[254] Zelinka, I., Vasek, V., Kolomaznik, K., Dostal, P., and Lampinen, J. Memetic algorithm and global optimization of chemical reactor. In *PC Control 2001, 13th International Conference on Process Control, High Tatras, Slovakia* (2001).

[255] Zwick, M., Lovell, B., and Marsh, J. Global optimization studies on the 1-d phase problem. *International Journal of General Systems 25*, 1 (1996), 47–59.