Journal of Heuristics, 8: 541–564, 2002
© 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

A Taxonomy of Hybrid Metaheuristics

E.-G. TALBI
Laboratoire d'Informatique Fondamentale de Lille, URA CNRS 369, Cité scientifique, 59655 Villeneuve d'Ascq Cedex, France
email: talbi@lifl.fr

Abstract

Hybrid metaheuristics have received considerable interest these recent years in the field of combinatorial optimization. A wide variety of hybrid approaches have been proposed in the literature. In this paper, a taxonomy of hybrid metaheuristics is presented in an attempt to provide a common terminology and classification mechanisms. The taxonomy, while presented in terms of metaheuristics, is also applicable to most types of heuristics and exact optimization algorithms.

As an illustration of the usefulness of the taxonomy an annoted bibliography is given which classifies a large number of hybrid approaches according to the taxonomy.

Key Words: taxonomy, combinatorial optimization, metaheuristics, hybrid algorithms, parallel algorithms

1. Introduction

Computing optimal solutions is computationally intractable for many combinatorial optimization problems, e.g., those known as NP-hard. In practice, we are usually satisfied with "good" solutions, which are obtained by metaheuristic algorithms. In addition to single-solution search algorithms such as descent local search (LS) (Papadimitriou and Steiglitz, 1982), greedy heuristic (GH) (Lawler, 1976), simulated annealing (SA) (Kirkpatrick, Gelatt, and Vecchi, 1983), tabu search (TS) (Glover, 1989), there is a growth interest in population-based metaheuristics. Those metaheuristics include evolutionary algorithms (EA: genetic algorithms (GA) (Holland, 1975), evolution strategies (ES) (Rechenberg, 1973), genetic programming (Koza, 1992), etc.), ant colonies (AC) (Colorni, Dorigo, and Maniezzo, 1991) scatter search (SS) (Glover, 1977), and so on. We refer the reader to Osman and Laporte (1996) and Reeves (1993) for good overviews of metaheuristics.

Over the last years, interest in hybrid metaheuristics has risen considerably among researchers in combinatorial optimization. The best results found for many practical or academic optimization problems are obtained by hybrid algorithms. Combinations of algorithms such as descent local search, simulated annealing, tabu search, and evolutionary algorithms have provided very powerful search algorithms.

In this paper, a taxonomy of hybrid algorithms is presented in an attempt to provide a common terminology and classification mechanisms. The goal of the general taxonomy given here is to provide a mechanism to allow comparison of hybrid algorithms in a qualitative way. In addition, it is hoped the categories and their relationships to each other have been chosen carefully enough to indicate areas in need of future work as well as to help classify future work. Among existing taxonomies in other domains, one can find examples of flat and hierarchical classifications schemes. The taxonomy proposed here is a combination of these two schemes—hierarchical as long as possible in order to reduce the total number of classes, and flat when the descriptors of the algorithms may be chosen in an arbitrary order.

In fact, the taxonomy could usefully be employed to classify any hybrid optimization algorithm (specific heuristics, exact algorithms). However, we shall focus our attention on hybrid metaheuristics since they are general heuristics applicable to a wide class of large optimization problems. We extend the basic classification by defining the space of hybrid metaheuristics as a grammar, where each sentence is a method that describes a combination of metaheuristics.

Some complementary works have been proposed in the literature to define a unified taxonomy of metaheuristics. In Calégari et al. (1999), the authors extract the fundamental ingredients of well known population-based metaheuristics, such as genetic algorithms, scatter search, evolutions strategies, and ant systems to define a unified taxonomy. The same approach has been proposed for local search metaheuristics in Vaessens, Aarts, and Lenstra (1992). In this paper, we propose a "high-level" description of hybrid metaheuristics. The internal working and the algorithmic aspects of a given metaheuristic are not considered. Indeed, a term like GA may cover many quite different algorithms nowadays.

The paper is organized as follows. First, we discuss design issues of hybrid metaheuristics. We present a taxonomy that tries to encompass all published work to date in the field and provide a unifying view of it. Then, we will focus on the implementation issues of hybrid metaheuristics. In Section 4, a grammar which generalizes the basic hybridization schemes is proposed. Finally, an annotated bibliography which classifies 124 references according to the taxonomy is given.

2. Design issues

Hybridization of heuristics involves a few major issues which may be classified as design and implementation. The former category concerns the hybrid algorithm itself, involving issues such as functionality and architecture of the algorithm. The implementation consideration includes the hardware platform, programming model and environment on which the algorithm is to run. In our paper, we try to make a difference between the design issues used to introduce hybridization and the implementation issues that depend on the execution model of the algorithms.

The taxonomy will be kept as small as possible by proceeding in a hierarchical fashion as long as possible, but some choices of characteristics may be made independent of previous design choices, and thus will be specified as a set of descriptors from which a subset may be chosen.

2.1. Hierarchical classification

The structure of the hierarchical portion of the taxonomy is shown in figure 1. A discussion about the hierarchical portion then follows.

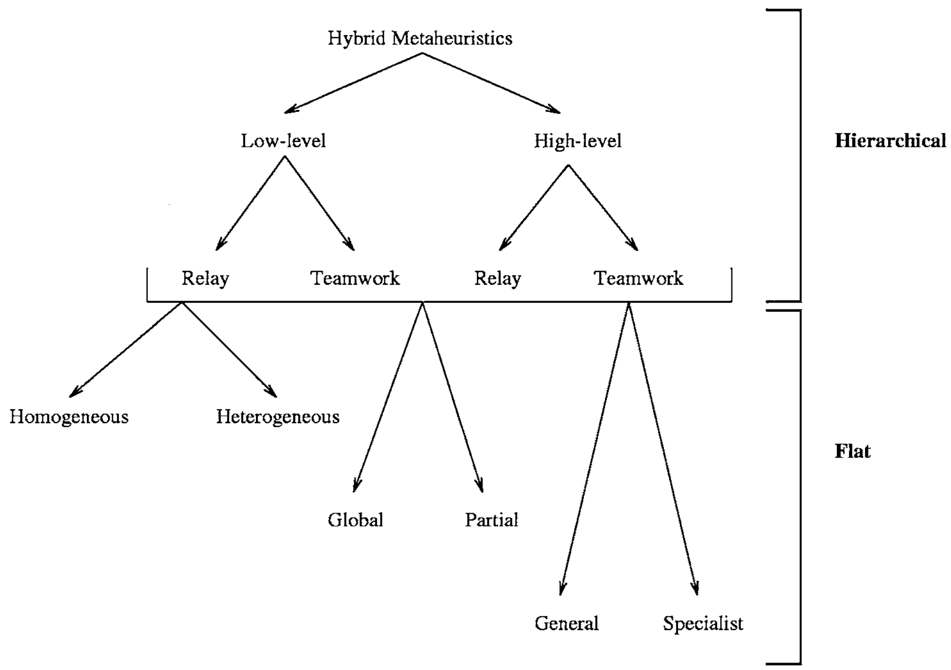

Figure 1. Classification of hybrid metaheuristics (design issues).

2.1.1. Low-level versus high-level. At the first level, we may distinguish between low-level and high-level hybridizations.

The low-level hybridization addresses the functional composition of a single optimization method. In this hybrid class, a given function of a metaheuristic is replaced by another metaheuristic.

In high-level hybrid algorithms, the different metaheuristics are self-contained. We have no direct relationship to the internal workings of a metaheuristic.

2.1.2. Relay versus teamwork. In relay hybridization, a set of metaheuristics is applied one after another, each using the output of the previous as its input, acting in a pipeline fashion.

Teamwork hybridization represents cooperative optimization models, in which we have many parallel cooperating agents, where each agent carries out a search in a solution space.

Four classes are derived from this hierarchical taxonomy:

2.1.2.1. LRH (Low-level relay hybrid). This class of hybrids represents algorithms in which a given metaheuristic is embedded into a single-solution metaheuristic. Few examples from the literature belong to this class.

O. Martin and S.W. Otto introduce in Martin, Otto, and Felten (1992) a LRH hybrid which combines simulated annealing with local search to solve the traveling salesman problem. The main idea is to embed deterministic local search techniques into simulated annealing

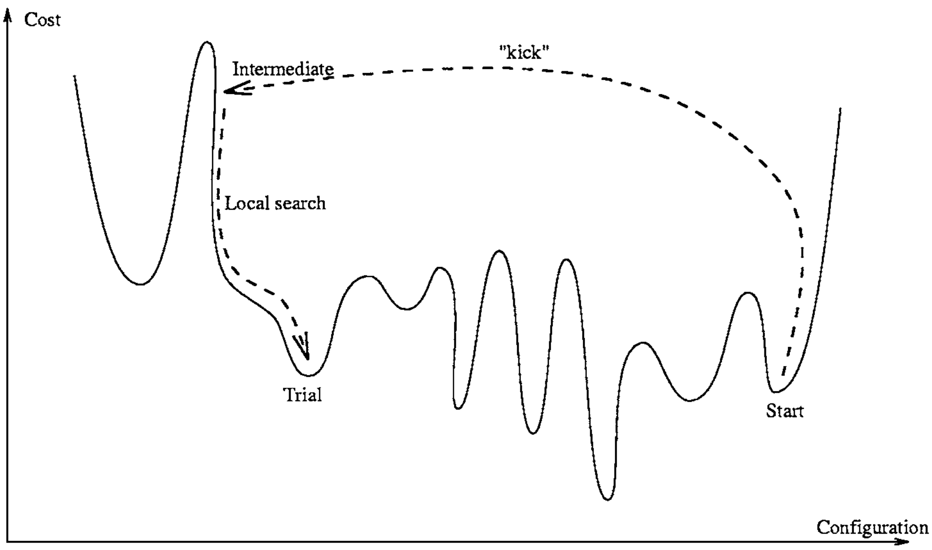

Figure 2. An example of LRH hybridization embedding local search in simulated annealing. The figure gives a schematic representation of the objective function and the configuration modification procedure used in the hybrid algorithm.

so that the Markov chain explores only local optima. The algorithm proceeds as follows: Suppose the configuration is currently locally optimal. This is labeled *Start* in figure 2. They apply a perturbation or a "kick" to this configuration which significantly changes *Start.* After the kick, we reach the configuration labeled *Intermediate* in the figure. Instead, they notice that is much better to first improve *Intermediate* by a local search and apply the accept/reject test of simulated annealing only afterwards. The local search takes us from *Intermediate* to the configuration labeled *Trial*, and then the accept/reject test is applied. If *Trial* is accepted, one has managed to find an interesting large change to *Start.* If *Trial* is rejected, return to *Start.* Many of the barriers (the "ridges") of the cost landscape are jumped over in one step by the hybrid metaheuristic.

To implement the above hybridization, the choice for an appropriate "kick" should be adapted to both the optimization problem and the local search method used. For the traveling salesman problem, if the local search algorithm used is the 3-opt local search heuristic, the "kick" move must apply a k-change with $k > 3$ to prevent cycles.

2.1.2.2. LTH (Low-level teamwork hybrid). Two competing goals govern the design of a metaheuristic: exploration and exploitation. Exploration is needed to ensure that every part of the space is searched enough to provide a reliable estimate of the global optimum. Exploitation is important since the refinement of the current solution will often produce a better solution. Population-based heuristics (genetic algorithms, scatter search, ant colonies, etc.) are powerful in the exploration of the search space, and weak in the exploitation of the solutions found.

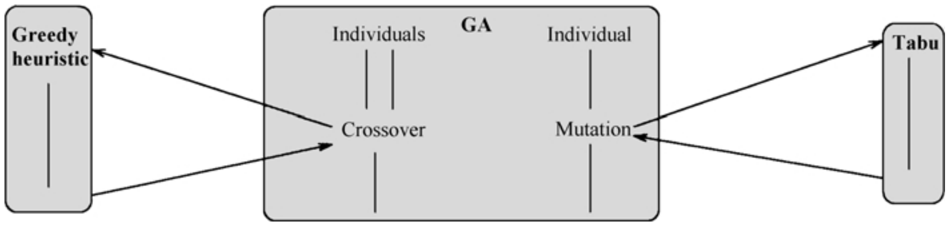

Figure 3. Low-level teamwork hybrid (LTH). For instance, a tabu search is used as a mutation operator and a greedy heuristic as a crossover operator in a genetic algorithm.

Therefore, most efficient population-based heuristics have been coupled with local search heuristics such as hill-climbing, simulated annealing and tabu search, which are powerful optimization methods in terms of exploitation. The two classes of algorithms have complementary strengths and weaknesses. The local search algorithms will try to optimize locally, while the population-based algorithms will try to optimize globally. In LTH hybrid, a metaheuristic is embedded into a population-based metaheuristic (figure 3).

For example, when a GA is used as a global optimizer, its standard operators may be augmented with the ability to perform local search. Instead of using a blind operator acting regardless of the fitness of the original individual and the operated one, we use an operator which is a heuristic that considers an individual as the origin of its search, applies itself, and finally replaces the original individual by the enhanced one. The use of local search with GAs is also inspired by biological models of learning and evolution. GAs take many cues from mechanisms observed in natural evolution. Similarly, models of learning are often equated with techniques for local optimization (Rumelhart, Hinton, and Williams, 1986). Research on the interaction between evolution and learning had naturally led computer scientists to consider interactions between evolutionary algorithms and local optimization (Belew, McInerny, and Schraudolph, 1991).

The genetic functions replaced or extended are generally mutation and crossover.

– Mutation: The local search algorithm may be a simple hill-climber (Suh and Van Gucht, 1987; Ulder et al., 1990; Jog, Suh, and Van Gucht, 1989), tabu search (Fleurent and Ferland, 1994a; Kim, Hayashi, and Nara, 1995; Thiel and Voss, 1994), or simulated annealing algorithm (Brown, Huntley, and Spillane, 1989; Chen and Flann, 1994; Wang et al., 1995). This kind of operators is qualified *lamarckian.*[1] In the lamarckian model, an individual is replaced by the local optima found, contrary to the baldwin model where the local optima is just used to evaluate the individual. In several occasions, LTH has provided better results than other methods on difficult problems. Good results have been obtained on the graph coloring problem hybridizing a genetic algorithm with a tabu search (Fleurant and Ferland, 1996). In Chu (1997), a LTH algorithm has been used for the multi-constraint knapsack problem and the set covering problem. A local search algorithm which utilises problem-specific knowledge is incorporated into the genetic operators. Questions concerning the best use of local search with a GA have been addressed in Hart (1994).

– Crossover: Classical crossover operators don't use any heuristic information about a specific application domain. Some researchers introduce heuristic crossover in order to account for problem-specific information (Grefenstette, 1987). Greedy heuristics for the crossover operator have shown to improve GAs results when applied to job-shop scheduling, set covering, and traveling salesman problems (Davis, 1985).

This hybrid model has also been used to enhance many population-based metaheuristics: ant colonies (Taillard and Gambardella, 1997; Stutzle and Hoos, 1997), genetic programming (O'Reilly and Oppacher, 1995), and scatter search (Cung et al., 1997). Local search has been introduced to intensify the search.

2.1.2.3. HRH (High-level relay hybrid). In HRH hybrid, self-contained metaheuristics are executed in a sequence. For example, it is well known that evolutionary algorithms are not well suited for fine-tuning structures which are very close to optimal solutions. Instead, the strenght of EAs is in quickly locating the high performance regions of vast and complex search spaces. Once those regions are located, it may be useful to apply local search heuristics to the high performance structures evolved by the EA.

A fundamental practical remark is that after a certain amount of time, the population is quite uniform and the fitness of the population is no longer decreasing. The odds to produce fitter individuals are very low. That is, the process has fallen into a basin of attraction from which it has a low probability to escape.

The exploitation of the already found basin of attraction to find as efficiently as possible the optimal point in the basin is recommended. It is experimentally clear that the exploitation of the basin of attraction that has been found may be more efficiently performed by another algorithm than by an EA. Hence, it is much more efficient to use a local search algorithm such as a hill-climbing or tabu search (see figure 4). The HRH hybridization may use a greedy heuristic to generate a good initial population for the EA (see figure 4).

Many authors have used the idea of HRH hybridization for EAs. In Mahfoud and Goldberg (1995) and Talbi, Muntean, and Samarandache (1994), the authors introduce respectively simulated annealing and tabu search to improve the population obtained by a GA. In Nissen

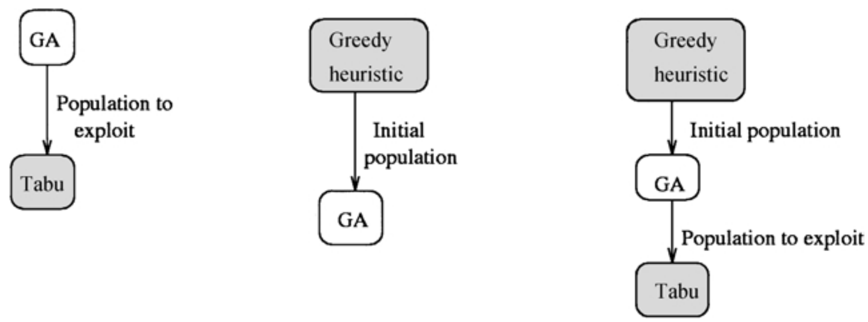

Figure 4. High-level relay hybridization. Three instances of this hybridization scheme are represented. There may be more than three algorithms to be pipelined.

(1994), the author introduces hill-climbing to improve the results obtained by an ES. In Lin, Kao, and Hsu (1991), the algorithm proposed starts from simulated annealing and uses GAs to enrich the solutions found. Experiments performed on the graph partitioning problem using the tabu search algorithm exploiting the result found by a GA give better results than a search performed either by the GA, or the tabu search alone (Talbi, Muntean, and Samarandache, 1994).

2.1.2.4. HTH (High-level teamwork hybrid). The HTH scheme involves several self-contained algorithms performing a search in parallel, and cooperating to find an optimum. Intuitively, HTH will ultimately perform at least as well as one algorithm alone, more often perform better, each algorithm providing information to the others to help them.

An example of HTH based on GAs is the island model. The population in this model is partitioned into small subpopulations by geographic isolation. A GA evolves each subpopulation and individuals can migrate between subpopulations (figure 5). This model is controlled by several parameters: the topology that defines the connections between subpopulations, the migration rate that controls the number of migrant individuals, the replacement strategy used, and a migration interval that affects how often migration occurs. In some island models, the individuals really migrate and therefore leaves empty space in the original population. In general, the migrated individuals remain in the original population (pollination model (Sprave, 1999)).

Tanese proposed a GA based HTH that used a 4-D hypercube topology to communicate individuals from one subpopulation to another (Tanese, 1987). Migrations occured at uniform periods of time between neighbor subpopulations along one dimension of the hypercube. The migrants were chosen probabilistically from the best individuals in the subpopulation and they replaced the worst individuals in the receiving subpopulation.

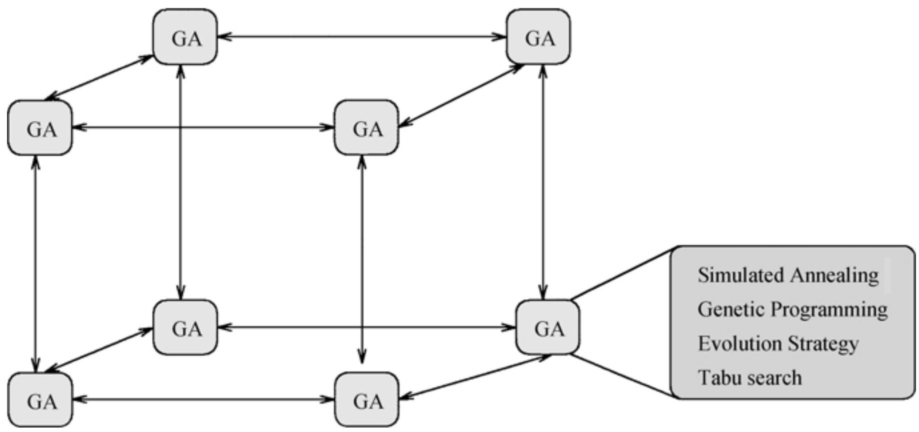

Figure 5. The island model of genetic algorithms as an example of high-level teamwork hybrid (HTH). The same model has been used with different topologies for simulated annealing, genetic programming, evolution strategies, and tabu search.

Cohoon et al. (1987) proposed a HTH based on the theory of "punctuated equilibria". A linear placement problem was used as a benchmark and experimented using a mesh topology. They found that the algorithm with migration outperformed the algorithm without migration and the standard GA. This work was later extended using a VLSI design problem (graph partitioning) on a 4-D hypercube topology (Cohoon, Martin, and Richards, 1990, 1991).

Belding in Belding (1995) attempted to extend the Tanese's work using the Royal Road functions. Migrants were sent to a random selected subpopulation, rather than using a hypercube topology. The global optimum was found more often when migration (with cooperation) was used than in completely isolated cases (without cooperation).

The HTH hybrid model has also been applied to simulated annealing (De Falco, Del Balio, and Tarantino, 1995), genetic programming (Koza and Andre, 1995), evolution strategies (Voigt, Born, and Santibanez-Koref, 1990), ant colonies (Mariano and Morales, 1998), scatter search (Cung et al., 1999) and tabu search (De Falco et al., 1994).

2.2. *Flat classification*

2.2.1. Homogeneous versus heterogeneous. In homogeneous hybrids, all the combined algorithms use the same metaheuristic. Hybrid algorithms such as the island model for GAs (Pettey, Leuze, and Grefenstette, 1987), belong to this class of hybrids. In general, different parameters are used for the algorithms. For example, in HTH based on tabu search, the algorithms may be initialized with different initial solutions, tabu list sizes, etc. (Voss, 1993).

In heterogeneous algorithms, different metaheuristics are used (figure 6). A heterogeneous HTH algorithm based on genetic algorithms and tabu search has been proposed in Crainic, Nguyen, and Gendreau (1997) to solve a network design problem. The population of the GA is asynchronously updated by multiple tabu search algorithms. The best solutions found by tabu search algorithms build an elite population for the GA.

The GRASP method (Greedy Randomized Adaptive Search Procedure) may be seen as an iterated heterogenous HRH hybrid, in which local search is repeated from a number of

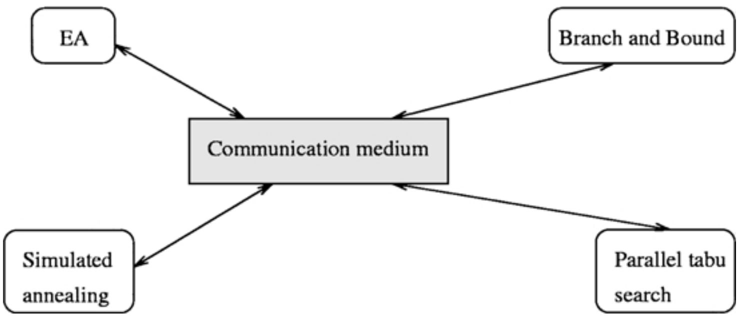

Figure 6. High-level teamwork hybridization HTH (heterogeneous, global, general). Several search algorithms cooperate, co-adapt, and co-evolve a solution.

initial solutions generated by randomized greedy heuristic (Feo, Resende, and Smith, 1994; Feo, Venkatraman, and Bard, 1991). The method is called adaptive because the greedy heuristic takes into account the decisions of the precedent iterations (Feo and Resende, 1995).

2.2.2. Global versus partial. From another point of view, we can also distinguish two kinds of cooperation: global and partial.

In global hybrids, all the algorithms search in the whole research space. The goal is here to explore the space more thoroughly. All the above mentioned hybrids are *global* hybrids, in the sense that all the algorithms solve the whole optimization problem. A global HTH algorithm based on tabu search has been proposed in Crainic, Toulouse, and Gendreau (1993), where each tabu search task performs a given number of iterations, then broadcasts the best solution. The best of all solutions becomes the initial solution for the next phase.

In partial hybrids, the problem to be solved is decomposed into sub-problems, each one having its own search space. Then, each algorithm is dedicated to the search in one of these sub-spaces. Generally speaking, the sub-problems are all linked with each others, thus involving constraints between optima found by each algorithm. Hence, the algorithms communicate in order to respect these constraints and build a global viable solution to the problem. This approach has been applied for simulated annealing and tabu search algorithms (Taillard, 1993).

An application of partial homogeneous HTH has been done for the job-shop scheduling problem (Husbands, Mill, and Warrington, 1990). The search algorithm is a GA. Each GA evolves individuals of a specie which represent the process plan for one job. Hence, there are as many cooperating GAs as there are jobs. The communication medium collects fitted individuals from each GA, and evaluates the resulting schedule as a whole, rewarding the best process plans.

2.2.3. Specialist versus general. All the above mentioned hybrids are *general* hybrids, in the sense that all the algorithms solve the same target optimization problem. *Specialist* hybrids combine algorithms which solve different problems. An example of such a HTH approach has been developed in Bachelet et al. (1998) to solve the quadratic assignment problem (QAP). A parallel tabu search is used to solve the QAP, while a genetic algorithm makes a diversification task, which is formulated as another optimization problem (figure 7). The frequency memory stores information relative to all the solutions visited during the tabu search. The genetic algorithm refers to the frequency memory to generate solutions being in unexplored regions.

Another approach of specialist hybrid HRH heuristics is to use a heuristic to optimize another heuristic, i.e. find the optimal values of the parameters of the heuristic. This approach has been used to optimize simulated annealing and noisy methods (NM) by GA (Krueger, 1993), ant colonies (AC) by GA (Abbattista, Abbattista, and Caponetti, 1995), and a GA by a GA (Shahookar and Mazumder, 1990). In Shahookar and Mazumder (1990), the three parameters optimized are the crossover rate, inversion rate, and mutation rate. The individuals in the population of the optimizer GA consist of three integers representing the mutation rate, inversion rate, and crossover rate. The fitness of an individual is taken

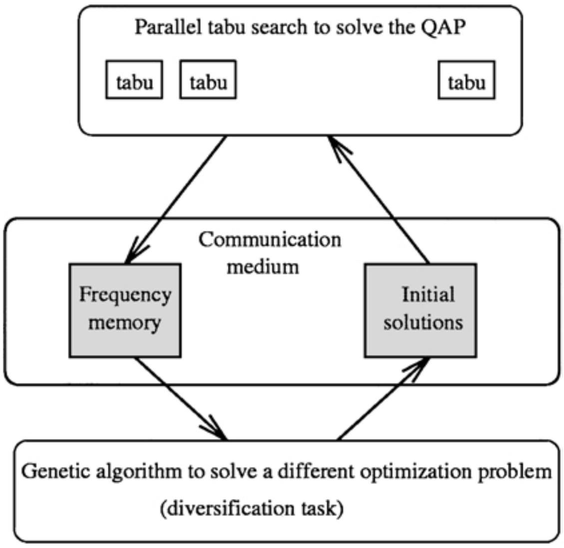

Figure 7. High-level teamwork hybridization HTH (global, heterogeneous, specialist). Several search algorithms solve different problems.

to be the fitness of the best solution that the GA can find in the entire run, using these parameters.

3. Implementation issues

The structure of the taxonomy concerning implementation issues is shown in figure 8. A discussion about this taxonomy then follows.

3.1. Specific versus general-purpose computers

Application specific computers differ from general purpose ones in that they usually only solve a small range of problems, but often at much higher rates and lower cost. Their internal structure is tailored for a particular problem, and thus can achieve much higher efficiency and hardware utilization than a processor which must handle a wide range of tasks.

In recent years, the advent of programmable logic devices has made easier to build specific computers for metaheuristics such as simulated annealing (Abramson, 1992) and genetic algorithms (Salami and Cain, 1996). In Abramson et al. (1997), a general architecture acting as a template for designing a number of specific machines for different metaheursitics (SA, TS, etc.) has been proposed. The processor is built with XILINX FPGAs and APTIX interconnexion chips. Experiments evaluating a simulated annealing algorithm to solve the traveling salesman problem achieved a speedup of about 37 times over an IBM RS6000 workstation. To our knowledge, this approach has not been yet proposed for hybrid metaheuristics.

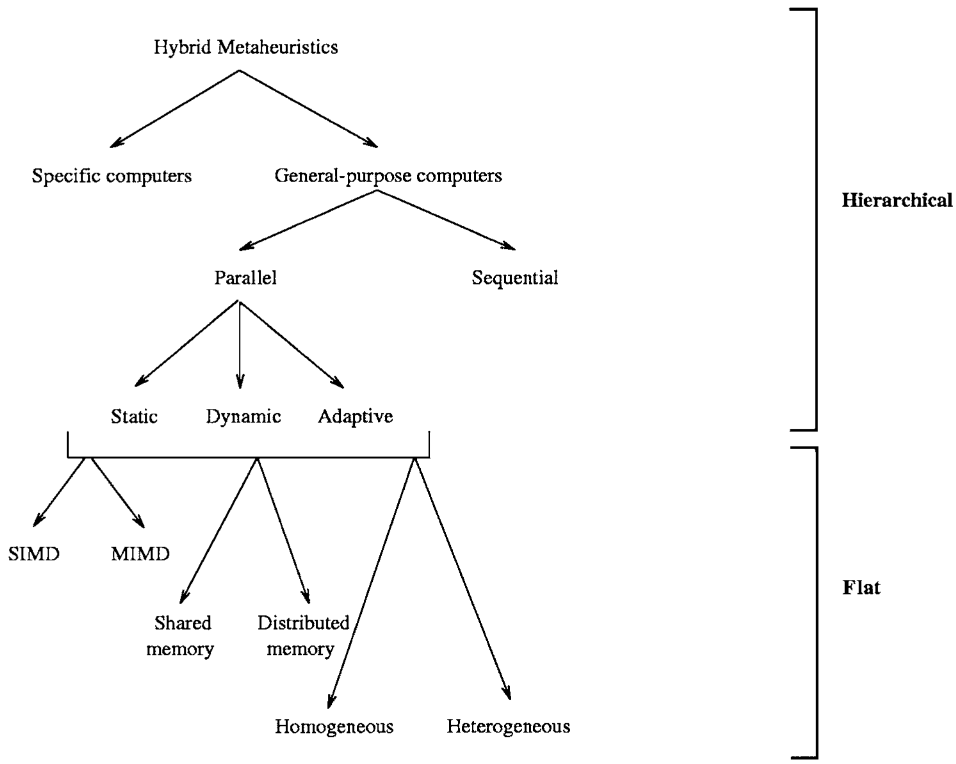

Figure 8. Classification of hybrid metaheuristics (implementation issues).

3.2. Sequential versus parallel

Most of the proposed hybrid metaheuristics are sequential programs. According to the size of problems, parallel implementations of hybrid algorithms have been considered. The easiness to use a parallel and distributed architecture has been acknowledged for the HTH hybrid model. Parallel hybrids may be classified using the different characteristics of the target parallel architecture:

- SIMD versus MIMD: In SIMD (Single Instruction stream, Multiple Data stream) parallel machines, the processors are restricted to execute the same program. They are very efficient in executing synchronized parallel algorithms that contain regular computations and regular data transfers. So, SIMD machines have been used for some parallel hybrid algorithms such a HTH based on tabu search arranged in a 2-dimentional cyclic mesh on a Maspar MPP-1 (De Falco, Del Balio, and Tarantino, 1994).

  When the computations or the data transfers become irregular or asynchronous, the SIMD machines become much less efficient. In parallel MIMD (Multiple Instruction stream, Multiple data stream), the processors are allowed to perform different types of instructions on different data. HTH hybrids based respectively on tabu search (De Falco et al.,

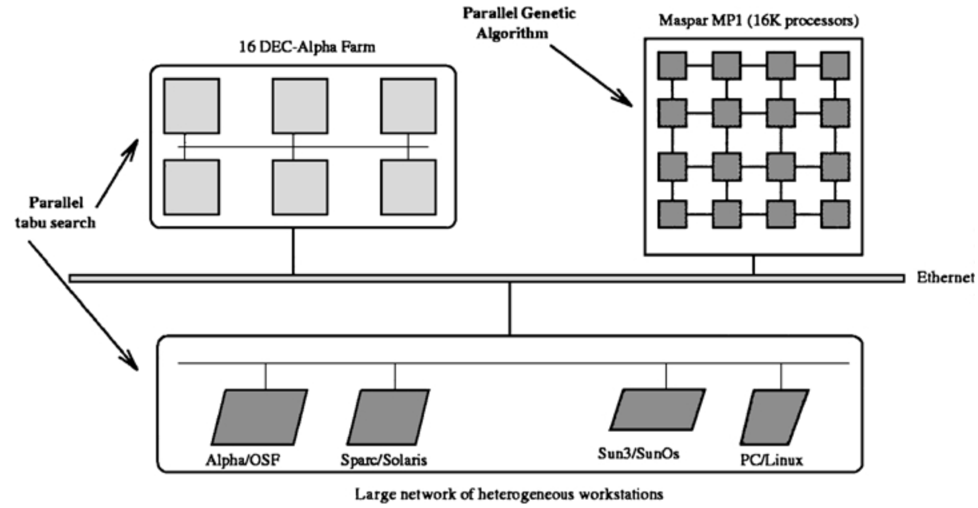

Figure 9. Parallel implementation of heterogeneous HTH algorithms.

1994; Fiechter, 1994), simulated annealing, and genetic algorithms (Muhlenbein, Schomisch, and Born, 1991) have been implemented on networks of transputers.

- Shared-memory versus Distributed-memory: The advantages of parallel hybrids implemented on shared-memory parallel architectures are their simplicity. However, parallel distributed-memory architectures offer a more flexible and fault-tolerant programming platform.
- Homogeneous versus Heterogeneous: Most massively parallel machines (MPP) and cluster of workstations (COW) such as IBM SP/2, Cray T3D, and DEC Alpha-farms are composed of homogeneous processors. The proliferation of powerful workstations and fast communication networks have shown the emergence of heterogeneous network of workstations (NOW) as platforms for high-performance computing.

We are experimenting with a specialist heterogeneous HTH to solve the QAP, implemented on a platform combining an MPP, a COW and a NOW (Bachelet, Preux, and Talbi, 1996). A GA is cooperating with multiple tabu search algorithms. The GA has been implemented on a massively parallel 16K processor Maspar MP-1 following a cellular synchronous model while the parallel tabu algorithms are running on a 16 DEC Alpha-farm and a large network of heterogeneous workstations (figure 9).

3.3. *Static, dynamic or adaptive*

Parallel heuristics fall into three categories depending on whether the number and/or the location of work (tasks, data) depend or not on the load state of the target parallel machine:

- *Static*: This category represents parallel heuristics in which both the number of tasks of the application and the location of work (tasks or data) are generated at compilation time (static scheduling). The allocation of processors to tasks (or data) remains unchanged during the execution of the application regardless of the current state of the parallel machine. Most of the proposed parallel heuristics belong to this class.

An example of such an approach for TS is presented in Porto and Ribeiro (1996). The neighborhood is partioned in equal size partitions depending on the number of workers, which is equal to the number of processors of the parallel machine. In Chakrapani and Skorin-Kapov (1993), the number of tasks generated depends on the size of the problem and is equal to n^2, where n is the problem size. In Bachelet, Preux, and Talbi (1996), a parallel GA is proposed, where the number of tasks generated is equal to the population size which is fixed at compilation time.

When there are noticeable load or power differences between processors, the search time of the static approach presented is derived by the maximum execution time over all processors (presumably on the most highly loaded processor or the least powerful processor). A significant number of tasks are often idle waiting for other tasks to complete their work.

- *Dynamic*: To improve the performance of parallel static heuristics, dynamic load balancing must be introduced (Badeau et al., 1995; Porto and Ribeiro, 1996). This class represents heuristics for which the number of tasks is fixed at compilation time, but the location of work (tasks, data) is determined and/or changed at run-time. Load balancing requirements are met in Porto and Ribeiro (1996) by a dynamic redistribution of work between processors. During the search, each time a task finishes its work, it proceeds to a work-demand.

  However, the degree of parallelism in this class of algorithms is not related to load variation in the target machine: when the number of tasks exceeds the number of idle nodes, multiple tasks are assigned to the same node. Moreover, when there are more idle nodes than tasks, some of them will not be used.
- *Adaptive*: *Parallel adaptive programs* are parallel computations with a dynamically changing set of tasks. Tasks may be created or killed as a function of the load state of the parallel machine. A task is created automatically when a node becomes idle. When a node becomes busy, the task is killed. In Bachelet et al. (1998), a parallel adaptive implementation has been proposed for HTH specialist hybrid combining tabu search and genetic algorithms.

4. A grammar for extended hybridization schemes

Given a set of optimization algorithms A_i, we have presented a classification of basic hybridizations, which can be described by the following notations:

- *LRH* $(A_1(A_2))$ (homogeneous, heterogeneous) (partial, global) (specialist, general): The metaheuristic A_2 is embedded into the single-solution metaheuristic A_1.
- *HRH* $(A_1 + A_2)$ (homogeneous, heterogeneous) (partial, global) (specialist, general): The self-contained metaheuristics A_1 and A_2 are executed in sequence.
- *LTH* $(A_1(A_2))$ (homogeneous, heterogeneous) (partial, global) (specialist, general): The metaheuristic A_2 is embedded into the population-based metaheuristic A_1.
- *HTH* (A_1, A_2) (homogeneous, heterogeneous) (partial, global) (specialist, general): The self-contained metaheuristics A_1 and A_2 are executed in parallel and cooperate.

$< hybrid - metaheuristic > \longrightarrow < design - issues >< implementation - issue >$
$< design - issues > \longrightarrow < hierarchical >< flat >$
$< hierarchical > \longrightarrow < LRH > \mid < LTH > \mid < HRH > \mid < HTH >$
$< LRH > \longrightarrow \mathrm{LRH}(< metaheuristic >(< metaheuristic >))$
$< LTH > \longrightarrow \mathrm{LTH}(< metaheuristic >(< metaheuristic >))$
$< HRH > \longrightarrow \mathrm{HRH}(< metaheuristic > + < metaheuristic >)$
$< HTH > \longrightarrow \mathrm{HTH}(< metaheuristic >)$
$< HTH > \longrightarrow \mathrm{HTH}(< metaheuristic >, < metaheuristic >)$
$< flat > \longrightarrow (< nature > , < optimization > , < function >)$
$< nature > \longrightarrow$ homogeneous | heterogeneous
$< optimization > \longrightarrow$ global | partial
$< function > \longrightarrow$ general | specialist
$< implementation - issue > \longrightarrow$ sequential | parallel $< scheduling >$
$< scheduling > \longrightarrow$ static | dynamic | adaptive
$< metaheuristic > \longrightarrow$ LS | TS | SA | GA | ES | GP | NN
$< metaheuristic > \longrightarrow$ GH | AC | SS | NM | CLP | $< hybrid - metaheuristic >$

Figure 10. A grammar for extended hybridization schemes.

The different metaheuristics A_i treated in our examples are: SA (Simulated Annealing) (Kirkpatrick, Gelatt, and Vecchi, 1983), GA (Genetic Algorithms) (Holland, 1975), ES (Evolution Strategies (Rechenberg, 1973), EP (Evolutionary programming), GP (Genetic Programming) (Koza, 1992), NN (Neural Networks), LS (Descent Local Search) (Papadimitriou and Steiglitz, 1982), TS (Tabu Search) (Glover, 1989), GH (Greedy Heuristic) (Lawler, 1976), AC (Ant Colonies) (Colorni, Dorigo, and Maniezzo, 1991), SS (Scatter Search) (Glover, 1977), NM (Noisy Method) (Charon and Hudry, 1993), and CLP (Constraing Logic Programming) (Hentenryck, 1989).

These hybridizations should be regarded as primitives that can be combined in different ways. The grammar given in figure 10 generalises the basic hybridization schemes. One of the practical importance of the grammar is to specify the hybrid heuristic to use, if we have a metaheuristic problem solving environment.

Boese, Kahng, and Muddu (1994) suggested an adaptive multi-start (AMS) approach, which may be seen as a HRH (LS + LTH(GA(LS))) scheme. First, AMS generates a set of random starting solutions and runs an LS algorithm for each solution to find corresponding local optima. Then, AMS generates new starting solutions by combining features of the T best local optima seen so far, with T being a parameter of the approach. This mechanism bears some resemblance to GAs, but differs in that many solutions (instead of just two) are used to generate the new starting solutions. New local optima are obtained by running the LS algorithm from these new starting solutions, and the process iterates until some stop criterion is met.

D. Levine has used a HTH(HRH(GH + LTH(GA(LS)))) hierarchical scheme in his Ph.D. to solve set partitioning problems. Efficient results have been obtained with a parallel static implementation in solving big sized problems in real world applications (airline crew scheduling) (Levine, 1994). At the first level, a HTH hybrid based on the island model of parallel genetic algorithms is used. The initial population of each GA was generated by a greedy heuristic (the Chavatal heuristic (Chvatal, 1979), and a local search algorithm was used to improve the solutions at each generation of the GA.

The same hybrid scheme with a sequential implementation has been used in Braun (1990) to solve the traveling salesman problem. The local search algorithms used are the well known 2-opt and or-opt heuristics. The author reported some interesting results on the 442 and 666-city problems. He found the optimum of the 442-city problem, and a solution within 0.04% of the optimum for the 666-city problem.

A $\mathrm{HRH}(\mathrm{HRH}(\mathrm{GH}+\mathrm{LS})+\mathrm{LTH}(\mathrm{GA}(\mathrm{HRH}(\mathrm{GH}+\mathrm{LS}))))$ scheme is presented in Freisleben and Merz (1996) for the traveling salesman problem. The basic idea is to use a simple nearest-neighbor tour construction for creating the initial population of a GA, apply a local search algorithm to this initial population for producing a population of local optima, and let a GA operate on the search space of local optima to determine the global optimum.

5. Conclusion

A taxonomy for hybrid metaheuristics has been presented. It considers solutions to design and implementation issues. Treating the two problems orthogonally is beneficial because it allows one to study, understand, classify and evaluate the algorithms using a well defined set of criteria.

A high percentage of metaheuristics hybridizing population-based metaheuristics with local search heuristics has been proposed for various optimization problems. Pure population-based heuristics such as genetic algorithms, genetic programming, evolution strategies, scatter search, and ant colonies are not well suited to fine-tuned search in highly combinatorial spaces.

Most of the hybrid algorithms are sequential. The authors of those sequential approaches often indicate in their future work the parallelization of the algorithms. This is an indication of the growing interest in developing parallel hybrid algorithms. Parallel schemes ideally provide novel ways to parallelize hybrid algorithms by providing parallel models of the algorithms. Hence, instead of merely parallelizing and finely tuning a sequential algorithm which has, though important, however limited capabilities to be parallelized, parallel hybrids are inherently suited to parallel computer environments. Furthermore, it should be pointed out and emphasized that the HTH proposes natural ways to efficiently implement algorithms on heterogeneous computer environments which is recognized as a very tough problem today.

The utility of the proposed taxonomy was demonstrated by applying it to classify some solutions proposed in the literature. As is the case of any bibliography, there are many papers to be considered. The exclusion of any particular results has not been neither intentional nor should it be considered as a judgment of the merit of that work.

6. Annotated bibliography

In this section, examples will be taken from the published literature to demonstrate their relationships to one another with respect to our taxonomy. The purpose of this section is to show that many different hybrid algorithms can fit into the taxonomy. Although considerable effort was devoted to collect many references, the list is certainly not complete. However, the underlying bibliography will be continuously updated.

Reference	Design	Implementation	Optimization problem
Aarts et al. (1986)	HTH(SA) (hom,glo,gen)	Parallel static	VLSI design
Abbattista, Abbattista, and Caponetti (1995)	HRH(GA + AC) (het,glo,spe)	Sequential	Traveling salesman
Ackley (1987)	LTH (GA(LS)) (het,glo,gen)	Sequential	Function optimization
Adamidis and Petridis (1996)	HTH(GA) (hom,glo,gen)	Parallel static	Neural network design
Andreatta and Ribeiro (1994)	HTH(TS) (hom,glo,gen)	Sequential	Graph partitioning
Areibi and Vannelli (1994)	HRH(GA + TS) (het,glo,gen)	Sequential	Circuit partitioning VLSI design
Asveren and Molitor (1996)	HTH(LTH(GA(LS)) (het,glo,gen)) (hom,glo,gen)	Parallel	Traveling salesman
Badeau et al. (1997)	HTH(TS) (hom,glo,gen)	Parallel dynamic	Vehicle routing
Banerjee and Jones (1986)	HTH(SA) (hom,par,gen)	Parallel static	Placement of macro-cells
Becker and Le Cun (1988)	HRH(GH + NN) (het,glo,gen)	Sequential	Learning process
Bianchini and Brown (1992)	HTH(GA) (hom,glo,gen)	Parallel static	0-1 integer linear programming
Boese, Kahng, and Muddu (1994)	HRH(LS + LTH(GA(LS)) (het,glo,gen)) (het,glo,gen))	Sequential	Traveling salesman, graph partitioning
Bohnenberger, Hesser, and Manner (1995)	LTH(GA(ES)) (het,glo,spe)	Sequential	Design of truss structures
Brown, Huntley, and Spillane (1989)	LTH(GA(SA)) (het,glo,gen)	Parallel static	Quadratic assignment
Bruns (1995)	LTH(GA(CLP))(het,glo,gen)	Sequential	Job-shop scheduling
Bui and Moon (1994)	LTH(GA(LS)) (het,glo,gen)	Sequential	Graph partitioning
Casotto, Romeo, and Sangiovanni-Vincentelli (1986)	HTH(SA) (hom,par,gen)	Parallel static	Placement of macro-cells
Chak and Feng (1995)	LTH(GA(LS)) (het,glo,gen)	Sequential	Function optimization
Chakrapani and Skorin-Kapov (1993)	LTH(GA(LS)) (het,glo,gen)	Parallel static	Quadratic assignment
Chen and Flann (1994)	LTH(GA(SA)) (het,glo,gen)	Sequential	Traveling salesman, function optimization
Cohoon et al. (1991)	HTH(GA) (hom,glo,gen)	Parallel static	Placement problem (Floorplan design)
Crainic and Gendreau (1997)	HTH(TS) (hom,glo,gen)	Parallel static	Network design
Crainic, Toulouse, and Gendreau (1995)	HTH(TS) (hom,par,gen)	Parallel static	Multi-commodity location-allocation
Cung et al. (1997)	LTH(SS(TS)) (het,glo,gen)	Sequential	Quadratic assignment
Davis (1985)	LTH(GA(GH)) (het,glo,gen)	Sequential	Job-shop scheduling
Dozier, Bowen, and Bahler (1995)	LTH(GA(LS)) (het,glo,gen)	Sequential	Constraint satisfaction
Duvivier, Preux, and Talbi (1996)	LTH (GA(LS)) (het,glo,gen)	Sequential	Job-shop scheduling
East and Rowe (1996)	HTH(GA) (hom,glo,gen)	Sequential	Binary function optimization
Fahlman (1988)	HRH (GH + NN) (het,glo,gen)	Sequential	Learning process
De Falco, Del Balio, and Tarantino (1994)	HTH(TS) (hom,glo,gen)	Parallel static	Mapping
Fleurant and Ferland (1996)	LTH(GA(HRH(GH + TS) (het,glo,gen)) (het,glo,gen)	Sequential	Graph coloring
Fleurent and Ferland (1994a)	LTH(GA(TS)) (het,glo,gen)	Sequential	Quadratic assignment
Fleurent and Ferland (1994b)	LTH(GA(HRH(GH + TS) (het,glo,gen)) (het,glo,gen)	Sequential	Maximum clique graph coloring satisfiability

Foo (1991)	HTH(LTH(GA(SA)) (het,glo,gen)) (hom,glo,gen)	Parallel static	Task allocation
Fourman (1985)	LTH (GA(GH)) (het,glo,gen)	Sequential	Circuit layout
Freisleben and Merz (1996)	HRH(HRH(GH + LS) (het,glo,gen) + LTH(GA(HRH(GH + LS)) (het,glo,gen)) (het,glo,gen)) (het,glo,gen)	Sequential	Traveling salesman
Gorges-Schleuter (1989)	LTH (GA(LS)) (het,glo,gen)	Parallel static	Traveling salesman
Grefenstette (1987)	HRH (GH + LTH(GA(LS)) (het,glo,gen)) (het,glo,gen)	Sequential	Traveling salesman
Hancock and Smith (1990)	HRH(GA + NN) (het,glo,spe)	Sequential	(Learning face recognition)
Heijligers and Jess (1995)	LTH(GA(GH)) (het,glo,gen)	Sequential	High-level synthesis scheduling
Huntley and Brown (1991a)	LTH(GA(LS)) (het,glo,gen)	Parallel static	Quadratic assignment
Huntley and Brown (1991b)	LTH(GA(SA)) (het,glo,gen)	Parallel static	Quadratic assignement
Iba (1996)	HTH(GP,HTH(GP) (hom,par,gen)) (het,glo,gen)	Sequential	Tile world (multi-agent test bed)
Ishibuchi and Murata (1996)	LTH(GA(LS) (het,glo,gen)	Sequential	Flow-shop scheduling
Jog, Suh, and Van Gucht (1989)	LTH (GA(LS)) (het,glo,gen)	Sequential	Traveling salesman
Kim, Hayashi, and Nara (1995)	LTH(GA(HRH(SA + TS) (het,glo,spec)) (het,glo,gen)	Sequential	Maintenance scheduling
Kirkpatrick and Toulouse (1985)	HRH(HRH(GH + SA) (het,glo,gen) + LS) (het,glo,gen)	Sequential	Traveling salesman
Koza and Andre (1995)	HTH(GP) (hom,glo,gen)	Parallel static	Boolean function optimization
Kragelund (1997)	LTH(GA(LS)) (het,glo,gen)	Sequential	Timetabling
Kroger, Schwenderling, and Vornberger (1990)	LTH(GA(GH)) (het,glo,gen)	Parallel static	Bin-packing
Kroger, Schwenderling, and Vornberger (1991)	HRH(GH + HTH(GA) (hom,glo,gen)) (het,glo,gen)	Parallel static	Packing
Krueger (1993)	HRH(GA + SA) (het,glo,spe)	Parallel static	Traveling salesman 3-SAT
	HRH(GA + NM) (het,glo,spe)	Parallel static	Traveling salesman
Von Laszewski and Muhlenbein (1990)	LTH(GA(LS)) (het,glo,gen)	Parallel static	Graph partitioning
Lee and Lee (1992)	HTH(SA) (hom,glo,gen)	Parallel static	Graph partitioning
Levine (1994)	HTH(HRH(GH + LTH(GA(LS)) (het,glo,gen)) (het,glo,gen) (hom,glo,gen)	Parallel static	Set partitioning
Liepins and Hilliard (1987)	LTH(GA(GH)) (het,glo,gen)	Sequential	Set covering, job scheduling, traveling salesman
Lohmann (1990)	HTH(ES) (hom,glo,gen)	Sequential	Filter problem
Malek et al. (1989)	HTH(TS) (hom,glo,gen)	Parallel static	Traveling salesman
Martin and Otto (1996)	HTH(LRH(SA(LS)) (het,glo,gen)) (hom,glo,gen)	Parallel dynamic	Traveling salesman, graph partitioning
Martin, Otto, and Felten (1992)	LRH(SA(LS)) (het,glo,gen)	Sequential	Traveling salesman

(*Continued on next page.*)

(*Continued*).

Reference	Design	Implementation	Optimization problem
Muhlenbein, Georges-Schleuter, and Kramer (1998)	LTH(GA(LS)) (het,glo,gen)	Parallel static	Traveling salesman
Muhlenbein, Schomisch, and Born (1991)	HTH(LTH(GA(LS)) (het,glo,gen)) (hom,glo,gen)	Parallel static	Function optimization
Niar and Freville (1997)	HTH(TS) (hom,glo,gen)	Parallel static	0-1 multidimenstional knapsack
Nissen (1994)	HRH(ES + LS) (het,glo,gen)	Sequential	Quadratic assignment
O'Reilly and Oppacher (1995)	LTH(GP(LS)) (het,glo,gen)	Sequential	Function optimization
Pettey, Leuze, and Grefenstette (1987)	HTH(GA) (hom,glo,gen)	Parallel static	Function optimization
Piramuthu (1990)	HRH(GH + NN) (het,glo,gen)	Sequential	Learning process
Porto and Ribeiro (1996)	HTH(TS) (hom,par,gen)	Parallel dynamic	Task scheduling
Potter et al. (1992)	LTH(GA(LS)) (het,glo,gen)	Sequential	Multiple fault diagnostic
Potter, Miller, and Weyrich (1990)	HRH(GA + LS) (het,glo,gen)	Sequential	Multiple fault diagnostic
Rego and Roucairol (1996)	HTH(TS) (hom,glo,gen)	Parallel static	Vehicle routing
Rose et al. (1986)	HRH(LS + HTH(SA) (hom,par,gen)) (het,glo,gen)	Parallel static	VLSI placement
Rudolph (1990)	HTH(ES) (hom,glo,gen)	Parallel static	Traveling salesman
Shahookar and Mazumder (1990)	HRH(GA + GA) (hom,glo,spec)	Sequential	Cell placement
Shoukry and Aboutabl (1996)	HRH(NN + NN) (hom,glo,spec)	Sequential	Maximal common subgraph
Sloot, Kandorp, and Schoneveld (1995)	HTH(SA) (hom,glo,gen)	Parallel static	Traveling salesman
Stutzle and Hoos (1997)	LTH(AC(LS)) (hom,glo,gen)	Sequential	Traveling salemsman
Sun and Wan (1995)	HTH(GA) (hom,glo,gen)	Sequential	Function optimization
Taillard (1993)	HTH(TS) (hom,par,gen)	Parallel static	Vehicle routing
Taillard and Gambardella (1997)	LTH(AC(LS)) (het,glo,gen)	Sequential	Quadratic assignment
Tanese (1987)	HTH(GA) (hom,glo,gen)	Parallel static	Walsh function, optimization
Thiel and Voss (1994)	HRH(GH + LTH(GA(TS)) (het,glo,gen)) (het,glo,gen)	Sequential	Multiconstraint, 0-1 knapsack
Voigt, Born, and Santibanez-Koref (1990)	HTH(ES) (hom,glo,gen)	Parallel static	Function optimization
Yang et al. (1995)	HRH(LS + LTH(GA(LS)) (het,glo,gen)) (het,glo,gen)	Sequential	Microwave inverse scattering

Note

1. The name is an allusion to Jean Batiste de Lamarck's contention that phenotypic characteristics acquired during lifetime can become heritable traits.

References

Aarts, E.H.L., F.M.I. De Bont, J.H.A. Habers, and P.J.M. Van Laarhoven. (1986). "Parallel Implementations of the Statistical Cooling Algorithms." *Integration* 4, 209–238.

Abbattista, F., N. Abbattista, and L. Caponetti. (1995). "An Evolutionary and Cooperative Agent Model for Optimization." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 668–671.

Abramson, D., P. Logothetis, A. Postula, and M. Randall. (1997). "Application Specific Computers for Combinatorial Optimisation." In *The Australien Computer Architecture Workshop*, Sydney, Australia.

Abramson, D.A. (1992). "A Very High Speed Architecture to Support Simulated Annealing." *IEEE Computer* 25, 27–34.

Ackley, D.H. (1987). *A Connectionist Machine for Genetic Hillclimbing*. Boston, USA: Kluwer Academic Pub.

Adamidis, P. and V. Petridis. (1996). "Co-Operating Populations with Different Evolution Behaviours." In *IEEE Int. Conf. on Evolutionary Computation, ICEC'96*, Nagoya, Japan, pp. 188–191.

Andreatta, A.A. and C.C. Ribeiro. (1994). "A Graph Partitioning Heuristic for the Parallel Pseudo-Exhaustive Logical Test of VLSI Combinatorial Circuits." *Operations Research* 50, 1–36.

Areibi, S. and A. Vannelli. (1994). "Advanced Search Techniques for Circuit Partitioning." *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 16, 77–97.

Asveren, T. and P. Molitor. (1996). "New Crossover Methods for Sequencing Problems." In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schewefel (eds.), *Parallel Problem Solving from Nature PPSN4*, Vol. 1141 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 290–299.

Bachelet, V., Z. Hafidi, P. Preux, and E.-G. Talbi. (1998). "Diversifying Tabu Search by Genetic Algorithms." In *INFORMS'98 on Operations Research and Management Sciences Meeting*, Montréal, Canada.

Bachelet, V., P. Preux, and E.-G. Talbi. (1996). "Parallel Hybrid Meta-heuristics: Application to the Quadratic Assignment Problem." In *Parallel Optimization Colloquium POC96*, Versailles, France, pp. 233–242.

Badeau, P., M. Gendreau, F. Guertin, J.-Y. Potvin, and E. Taillard. (1995). "A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows." *RR CRT-95-84*, Centre de Recherche sur les Transports, Universitè de Montréal, Canada.

Badeau, P., F. Guertin, M. Gendreau, J.-Y. Potvin, and E.D. Taillard. (1997). "A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows." *Transportation Research* 5(2), 109–122.

Banerjee, P. and M. Jones. (1986). "A Parallel Simulated Annealing Algorithm for Standard Cell Placement on a Hypercube Computer." In *IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, California, USA, pp. 34–37.

Becker, S. and Y. Le Cun. (1988). "Improving the Convergence of Back-Propagation Learning with Second-Order Methods." In D. Touretzky, G. Hinton, and T. Sejnowski (eds.), *Connectionist Models Summer School*, Pittsburgh, USA, Morgan Kaufmann.

Belding, T. (1995). "The Distributed Genetic Algorithm Revisted." In D. Eshelmann (ed.), *Sixth Int. Conf. on Genetic Algorithms*. San Mateo, CA, Morgan Kaufmann.

Belew, R.K., J. McInerny, and N.N. Schraudolph. (1991). "Evolving Networks: Using Genetic Algorithms with Connectionist Learning." In C.G. Langton, C. Taylor, J.D. Doyne Farmer, and S. Rasmussen (eds.), *Second Conf. on Artificial Life*, Addison-Wesley, USA, pp. 511–548.

Bianchini, R. and C. Brown. (1992). "Parallel Genetic Algorithms on Distributed-Memory Architecturers," Technical Report 436, University of Rochester, Rochester, NY, USA.

Boese, K.D., A.B. Kahng, and S. Muddu. (1994). "New Adaptive Multi-Start Techniques for Combinatorial Global Optimizations." *Operation Research Letters* 16(2), 101–113.

Bohnenberger, O., J. Hesser, and R. Manner. (1995). "Automatic Design of Truss Structures Using Evolutionary Algorithms" In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 143–147.

Braun, H. (1990). "On Solving Traveling Salesman Problems by Genetic Algorithms." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*, Vol 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 129–133.

Brown, D.E., C.L. Huntley, and A.R. Spillane. (1989). "A Parallel Genetic Heuristic for the Quadratic Assignment Problem." In *Third Int. Conf. on Genetic Algorithms ICGA'89*, San Mateo, California, USA: Morgan Kauffmann, pp. 406–415.

Bruns, R. (1995). "Integration of Constraint Solving Techniques in Genetic Algorithms." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 33–38.

Bui,T.N. and B.R. Moon. (1994). "A Genetic Algorithm for a Special Class of the Quadratic Assignment Problem." *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Special Issue on Quadratic Assignment and Related Problems* 16, 99–116.

Calégari, P., G. Coray, A. Hertz, D. Kobler, and P. Kuonen. (1999). "A Taxonomy of Evolutionary Algorithms in Combinatorial Optimization." *Journal of Heuristics* 5(2), 145–158.

Casotto, A., F. Romeo, and A.L. Sangiovanni-Vincentelli. (1986). "A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells." In *IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, California, USA, pp. 30–33.

Chak, C.K. and G. Feng. (1995). "Accelerated Genetic Algorithms: Combined with Local Search Techniques for Fast and Accurate Global Search." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 378–383.

Chakrapani, J. and J. Skorin-Kapov. (1993). "Massively Parallel Tabu Search for the Quadratic Assignment Problem." *Annals of Operations Research* 41, 327–341.

Charon, I. and O. Hudry. (1993). "The Noising Method: A New Method for Combinatorial Optimization." *Operations Research Letters* 14, 133–137.

Chen, H. and N.S. Flann. (1994). "Parallel Simulated Annealing and Genetic Algorithms: A Space of Hybrid Methods." In Y. Davidor, H.-P. Schwefel, and R. Manner (eds.), *Third Conf. on Parallel Problem Solving from Nature*. Jerusalem, Israel, Berlin: Springer-Verlag, pp. 428–436.

Chu, P.C. (1997). "A Genetic Algorithm Approach for Combinatorial Optimization Problems." PhD Thesis, University of London, London, UK.

Chvatal, V. (1979). "A Greedy Heuristic for the Set Covering Problem." *Mathematics of Operations Research* 4(3), 233–235.

Cohoon, J., S. Hedge, W. Martin, and D. Richards. (1987). "Punctuated Equilibria: A Parallel Genetic Algorithm" In J.J. Grefenstette (ed.), *Second Int. Conf. on Genetic Algorithms*. Cambridge, MA, USA: MIT, pp. 148–154.

Cohoon, J., S. Hedge, W. Martin, and D. Richards. (1991). "Distributed Genetic Algorithms for the Floorplan Design Problem." *IEEE Trans. on Computer-Aided Design* 10(4), 483–492.

Cohoon, J.P., W.N. Martin, and D.S. Richards. (1990). "Genetic Algorithms and Punctuated Equilibria." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*. Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 134–141.

Cohoon, J.P., W.N. Martin, and D.S. Richards. (1991). "A Multi-Population Genetic Algorithm for Solving the k-Partition Problem on Hypercubes." In R.K. Belew and L.B. Booker (eds.), *Fourth Int. Conf. on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, pp. 244–248.

Colorni, A., M. Dorigo, and V. Maniezzo. (1991). "Distributed Optimization by Ant Colonies." In *European Conf. on Artificial Life*. Elsevier Publishing, pp. 134–142.

Crainic, T.D., M. Toulouse, and M. Gendreau. (1993). "Towards a Taxonomy of Parallel Tabu Search Algoithms." Technical Report CRT-933, Centre de Recherche sur les Transports, Université de Montréal, Montréal, Canada.

Crainic, T.G., and M. Gendreau. (1997). "A Cooperative Parallel Tabu Search for Capacitated Network Design," Technical Report CRT-97-27, Centre de recherche sur les transports, Université de Montréal, Montréal, Canada.

Crainic, T.G., A.T. Nguyen, and M. Gendreau. (1997). "Cooperative Multi-Thread Parallel Tabu Search with Evolutionary Adaptive Memory." In *2nd Int. Conf. on Metaheuristics*, Sophia Antipolis, France.

Crainic, T.G., M. Toulouse, and M. Gendreau. (1995). "Synchronous Tabu Search Parallelization Strategies for Multi-Commodity Location-Allocation with Balancing Requirements." *OR Spektrum*, 17, 113–123.

Cung, V-D., T. Mautor, P. Michelon, and A. Tavares. (1997). "A Scatter Search Based Approach for the Quadratic Assignment Problem." In *IEEE Int. Conf. on Evolutionary Computation ICEC'97*, Indianapolis, USA.

Cung, V.-D., T. Mautor, P. Michelon, and A. Tavares. (1999). "Recherche Dispersée Parallèle." In *Deuxième Congrés de la Socièté Francaise de Recherche Opératinnelle et d'Aide à la décision ROADEF'99*, Autrans, France.

Davis, L. (1985). "Job-Shop Scheduling with Genetic Algorithms." In J.J. Grefenstette (ed.), *Int. Conf. on Genetic Algorithms and their Applications*, Pittsburgh, pp. 136–140.

Dozier, G., J. Bowen, and D. Bahler. (1995). "Solving Randomly Generated Constraint Satisfaction Problems Using a Micro-Evolutionary Hybrid that Evolves a Population of Hill-Climbers." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 614–619.

Duvivier, D., P. Preux, and E.G. Talbi. (1996). "Climbing up NP-Hard Hills." In *The Fourth Int. Conf. on Parallel Problem Solving From Nature.* Berlin, Germany: Springer-Verlag, LNCS No. 1141, pp. 574–583.

East, I.R. and J. Rowe. (1996). "Effects of Isolation in a Distributed Population Genetic Algorithm." In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schewefel (eds.), *Parallel Problem Solving from Nature PPSN4.* Vol. 1141 of *LNCS*, Dortmund, Germany: Springer-Verlag, pp. 408–419.

Fahlman, S.E. (1988). "Faster-Learning Variations on Back-Propagation: An Empirical Study." In D. Touretzky, G. Hinton, and T. Sejnowsk (eds.), *Connectionist Models Summer School*, Pittsburgh, PA, USA, San Mateo, CA: Morgan Kaufmann, pp. 38–51.

De Falco, I., R. Del Balio, and E. Tarantino. (1994). "Solving the Mapping Problem by Parallel Tabu Search." In *IASTED Conf.*, Paris, France.

De Falco, I., R. Del Balio, and E. Tarantino. (1995). "An Analysis of Parallel Heuristics for Task Allocation in Multicomputers." *Computing* 3, 59.

De Falco, I., R. Del Balio, E. Tarantino, and R. Vaccaro. (1994). "Improving Search by Incorporating Evolution Principles in Parallel Tabu Search." In *Int. Conf. on Machine Learning*, pp. 823–828.

Feo, T.A. and M.G.C. Resende. (1995). "Greedy Randomized Adaptive Search Procedures." *Journal of Global Optimization* 6, 109–133.

Feo, T.A., M.G.C. Resende, and S.H. Smith. (1994). "A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set." *Operations Research* 42, 860–878.

Feo, T.A., K. Venkatraman, and J.F. Bard. (1991). "A GRASP for a Difficult Single Machine Scheduling Problem." *Computers and Operations Research* 18, 635–643.

Fiechter, C.-N. (1994). "A parallel Tabu Search Algorithm for Large Travelling Salesman Problems." *Discrete Applied Mathematics.*

Fleurent, C. and J.A. Ferland. (1994a). "Genetic Hybrids for the Quardratic Assignment Problem." *DI-MACS Series in Discrete Mathematics and Theoretical Computer Science* 16, 173–188.

Fleurent, C. and J.A. Ferland. (1994b). "Object-Oriented Implementation of Heuristic Search Methods for Graph Coloring, Maximum Clique, and Satisfiability." *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 26, 619–652.

Fleurant, C. and J.A. Ferland. (1996). "Genetic and Hybrid Algorithms for Graph Coloring." *Annals of Operations Research* 63.

Foo, S.-M. (1991). "An Approach of Combining Simulated Annealing and Genetic Algorithm." Master's Thesis, University of Illinois, Urbana-Champaign.

Fourman, M.P. (1985). "Compaction of Symbolic Layout Using Genetic Algorithms." In J.J. Grefenstette (ed.), *Int. Conf. on Genetic Algorithms and their Applications*, Pittsburgh, pp. 141–153.

Freisleben, B. and P. Merz. (1996). "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems." In *IEEE Int. Conf. on Evolutionary Computation, ICEC'96*, Nagoya, Japan, pp. 616–621.

Glover, F. (1977). "Heuristics for Integer Programming using Surrogate Constraints." *Decision Sciences* 8, 156–166.

Glover, F. (1989). "Tabu Search-Part I." *ORSA Journal of Computing* 1(3), 190–206.

Gorges-Schleuter, M. (1989). "Asparagos, an Asynchronous Parallel Genetic Optimization Strategy." In *3rd Int. Conf. Genetic Algorithms.* Morgan Kaufmann, USA. pp. 422–427.

Grefenstette, J.J., (1987). "Incorporating Problem Specific Knowledge into Genetic Algorithms." In L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, San Mateo, CA, USA: Morgan Kaufmann, pp. 42–60.

Hancock, P.J.B. and L.S. Smith. (1990). “Gannet: Genetic Design of a Neural Net for Face Recognition.” In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature.* Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 292–296.

Hart, W.E. (1994). “Adaptive Global Optimization with Local Search.” PhD Thesis, University of California, San Diego.

Heijligers, M.J.M. and J.A.G. Jess. (1995). “High-Level Synthesis Scheduling and Allocation Using Genetic Algorithms Based on Conctructive Topological Scheduling Techniques.” In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 56–61.

Hentenryck, V.P. (1989). *Constraint Satisfaction in Logic Programming.* MIT Press.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI, USA: Michigan Press University.

Huntley, C.L. and D.E. Brown. (1991a). “Parallel Genetic Algorithms with Local Search.” Technical Report IPC-TR-90-006, University of Virginia, Charlottesvile, VA, USA.

Huntley, C.L. and D.E. Brown. (1991b). “A Parallel Heuristic for Quadratic Assignment Problems.” *Computers and Operations Research* 18, 275–289.

Husbands, P., F. Mill, and S. Warrington. (1990). “Genetic Algorithms, Production Plan Optimisation and Scheduling.” In H.-P. Schewefel and R. Manner (eds.), *Parallel Problem Solving From Nature*, Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 80–84.

Iba, H. (1996). “Emergent Cooperation for Multiple Agents Using Genetic Programming.” In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schewefel (eds.), *Parallel Problem Solving from Nature PPSN4*, Vol. 1141 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 32–41.

Ishibuchi, H. and T. Murata. (1996). “Multi-Objective Genetic Local Search Algorithm.” In *IEEE Int. Conf. on Evolutionary Computation, ICEC'96*, Nagoya, Japan, pp. 119–124.

Jog, P., J.Y. Suh, and D. Van Gucht. (1989). “The Effects of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem.” In *3rd Int. Conf. Genetic Algorithms*, Morgan Kaufmann, USA.

Kim, H., Y. Hayashi, and K. Nara. (1995). “The Performance of Hybridized Algorithm of Genetic Algorithm Simulated Annealing and Tabu Search for Thermal Unit Maintenance Scheduling.” In *2nd IEEE Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 114–119.

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi. (1983). “Optimization by Simulated Annealing.” *Science* 220(4598), 671–680.

Kirkpatrick, S., and G. Toulouse. (1985). “Configuration Space Analysis of the Travelling Salesman Problem.” *J. Phys.* 46(1277).

Koza, J. and D. Andre. (1995). “Parallel Genetic Programming on a Network of Transputers.” Technical Report CS-TR-95-1542, Stanford University.

Koza, J.R. (1992). *Genetic Programming.* Cambridge, USA: MIT Press.

Kragelund, L.V. (1997). “Solving a Timetabling Problem Using Hybrid Genetic Algorithms.” *Software Practice and Experience* 27(10), 1121–1134.

Kroger, B., P. Schwenderling, and O. Vornberger. (1990). “Parallel Genetic Packing of Rectangles.” In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature.* Vol. 496 of *LNCS*, Dortmund, Germany. Springer-Verlag, pp. 160–164.

Kroger, B., P. Schwenderling, and O. Vornberger. (1991). “Genetic Packing of Rectangles on Transputers.” In P. Welch et al. (eds.), *Transputing 91.* IOS Press.

Krueger, M. (1993). “Méthodes d'analyse d'algorithmes d'optimisation stochastiques à l'aide d'algorithmes génétiques.” Ph.D. Thesis, Ecole Nationale Supèrieure des Télécommunications, Paris, France.

Lawler, E.L. (1976). *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, USA.

Lee, K.-G. and S.-Y. Lee. (1992). “Efficient Parallelization of Simulated Annealing Using Multiple Markov Chains: An Application to Graph Partitioning.” In T.N. Mudge (ed.), *Int. Conf. on Parallel Processing*. CRC Press, pp. 177–180.

Levine, D. (1994). “A Parallel Genetic Algorithm for the Set Partitioning Problem.” Ph.D. Thesis, Argonne National Laboratory, Illinois Institute of Technology, Argonne, USA.

Liepins, G.E. and M.R. Hilliard. (1987). “Greedy Genetics.” In *2nd Int. Conf. on Genetic Algorithms: Genetic Algorithms and Their Applications*, Hillsdale, NJ, USA, Lawrence Erlbaum, pp. 90–99.

Lin, F.T., C.Y. Kao, and C.C. Hsu. (1991). "Incorporating Genetic Algorithms into Simulated Annealing." *Proc. of the Fourth Int. Symp. on AI*. pp. 290–297.

Lohmann, R. (1990). "Application of Evolution Strategy in Parallel Populations." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*. Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 198–208.

Mahfoud, S.W. and D.E. Goldberg. (1995). "Parallel Recombinative Simulated Annealing: A Genetic Algorithm." *Parallel Computing* 21, 1–28.

Malek, M., M. Guruswamy, M. Pandya, and H. Owens. (1989). "Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem." *Annals of Operations Research* 21, 59–84.

Mariano, C.E. and E. Morales. (1998). "A Multiple Objective Ant-q Algorithm for the Design of Water Distribution Irrigation Networks." In *First International Workshop on Ant Colony Optimization ANTS'98*, Bruxelles, Belgique.

Martin, O.C. and S.W. Otto. (1996). "Combining Simulated Annealing with Local Search Heuristics." *Annals of Operations Research* 63, 57–75.

Martin, O.C., S.W. Otto, and E.W. Felten. (1992). "Large-Step Markoy Chains for the TSP: Incorporating Local Search Heuristics." *Operation Research Letters* 11, 219–224.

Muhlenbein, H., M. Georges-Schleuter, and M. Kramer. (1998). "Evolution Algorithms in Combinatorial Optimization." *Parallel Computing* 7, 65–85.

Muhlenbein, H., M. Schomisch, and J. Born. (1991). "The Parallel Genetic Algorithm as Function Optimizer." *Parallel Computing* 17, 619–632.

Niar, S. and A. Freville. (1997). "A Parallel Tabu Search Algorithm for the 0-1 Multidimensional Knapsack Problem." In *Int. Parallel Processing Symposium*, Geneva, Switzerland. IEEE Society.

Nissen, V. (1994). "Solving the Quadratic Assignment problem with Clues from Nature." *IEEE Transactions on Neural Networks* 5(1), 66–72.

O'Reilly, U.-M. and F. Oppacher. (1995). "Hybridized Crossover-Based Techniques for Program Discovery." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 573–578.

Osman, I.H. and G. Laporte. (1996). "Metaheuristics: A Bibliography." *Annals of Operations Research* 63, 513–628.

Papadimitriou, C.H. and K. Steiglitz. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.

Petty, C.B., M.R. Leuze, and J.J. Grefenstette. (1987). "A Parallel Genetic Algorithm." In *Proc. of the Second Int. Conf. on Genetic Algorithms*. Cambridge: MIT, pp. 155–161.

Piramuthu, S. (1990). "Feature Construction for Back-Propagation." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*, Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 264–268.

Porto, S.C.S. and C. Ribeiro. (1996). "Parallel Tabu Search Message-Passing Synchronous Strategies for Task Scheduling Under Precedence Constraints." *Journal of Heuristics* 1(2), 207–223.

Potter, W.D., J.A. Miller, B.E. Tonn, R.V. Gandham, and C.N. Lapena. (1992). "Improving the Reliability of Heuristic Multiple Fault Diagnosis via the Ec-based Genetic Algorithm." *Int. J. Artificial Intell., Neural Networks, Complex Problem-Solving Technol.* 2, 5–23.

Potter, W.D., J.A. Miller, and O.R. Weyrich. (1990). "A Comparison of Methods for Diagnostic Decision Making." *Expert Syst. Applicat. Int. J.* 1, 425–436.

Rechenberg, I. (1973). *Evolutionsstrategie : Optimierung technischer systeme nach prizipien der biologischen evolution*. Formann-Holzboog Verlag, Stuttgart, Germany.

Reeves, C.R. (1993). *Modern Heuristic Techniques for Combinatorial Problems*. Oxford, UK: Black Scientific Publications.

Rego, C. and C. Roucairol. (1996). "A Parallel Tabu Search Algorithm for the Vehicle Routing Problem." In I.H. Osman and J.P. Kelly (eds.), *Meta-Heuristics: Theory and Applications*. Kluwer, Norwell, MA, USA, pp. 253–295.

Rose, J.S., D.R. Blythe, W.M. Snelgrove, and Z.G. Vranecic. (1986). "Fast, High Quality VLSI Placement on a MIMD Multiprocessor." In *IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, pp. 42–45.

Rudolph, G. "Global Optimization by Means of Distributed Evolution Strategies." (1990). In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*. Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 209–213.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. (1986). "Learning Internal Representations by Error Propagation." In D.E. Rumelhart and J.L. McClelland (eds.), *Parallel Distributed Processing*, Vol. 1, MIT Press, USA, pp. 318–362.

Salami, M. and G. Cain. (1996). "Genetic Algorithm Processor on Reprogrammable Architectures." In *Fifth Annual Conference on Evolutionary Programming EP'96*, San Diego, California, USA, MIT Press.

Shahookar, K. and P. Mazumder. (1990). "A Genetic Approach to Standard Cell Placement Using Metagenetic Parameter Optimization." *IEEE Trans. Computer-Aided Design* 9(5), 500–511.

Shoukry, A. and M. Aboutabl. (1996). "Neural Network Approach for Solving the Maximal Common Subgraph Problem." *IEEE Trans. on Systems, Man, and Cybernetics* 26(5), 785–790.

Sloot, P.M., J.A. Kandorp, and A. Schoneveld. (1995). "Dynamic Complex Systems: A New Approach to Parallel Computing in Computational Physics." Technical Report TR-CS-95-08, University of Amsterdam, Netherlands.

Sprave, J. (1999). "A Unified Model of Non-panmictic Population Structures in Evolutionary Algorithms." In *Proc. of the 1999 Congress on Evolutionary Computation*, Vol. 2, Piscataway, NJ, IEEE Press, pp. 1384–1391.

Stutzle, T. and H.H. Hoos. (1997). "The MAX-MIN Ant System and Local Search for Combinatorial Optimization Problems: Towards Adaptive Tools for Global Optimization." In *2nd Int. Conf. on Metaheuristics*, Sophia Antipolis, France, INRIA, pp. 191–193.

Suh, J.Y. and D. Van Gucht. (1987). "Incorporating Heuristic Information into Genetic Search." In *2nd Int. Conf. Genetic Algorithms*, Lawrence Erlbaum Associates, USA, pp. 100–107.

Sun, Z. and Q. Wan. (1995). "A Modified Genetic Algorithm: Meta-leval Control of Migration in Distributed Ga." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 312–316.

Taillard, E. (1993). "Parallel Iterative Search Methods for Vehicle Routing Problem." *Networks* 23, 661–673.

Taillard, E.D. and L.M. Gambardella. (1997). "An Ant Approach for Structured Quadratic Assignment Problems." In C. Roucairol, I.H. Osman, S. Martello, and S. Voss (eds.), *2nd Int. Conf. on Metaheuristics*, Sophia Antipolis, France, INRIA, pp. 217–222.

Talbi, E.G., T. Muntean, and I. Samarandache. (1994). "Hybridation des Algorithmes Génétiques Avec la Recherche Tabou." In *Evolution Artificielle EA94*, Toulouse, France.

Tanese, R. (1987). "Parallel genetic Algorithms for a Hypercube." In *Proc. of the Second Int. Conf. on Genetic Algorithms*, MIT, Cambridge, MA, USA, pp. 177–183.

Thiel, J. and S. Voss. (1994). "Some Experiences on Solving Multiconstraint Zero-one Knapsack Problems with Genetic Algorithms." *INFOR* 32(4), 226–242.

Ulder, N.L.J., E.H.L. Aarts, H.-J. Bandelt, P.J.M. Van Laarhoven, and E. Pesch. (1990). "Genetic Local Search Algorithms for the Traveling Salesman Problem." In H.-P. Schewefel and R. Manner (eds.), *Parallel Problem Solving from Nature*. Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 109–116.

Vaessens, R., E. Aarts, and J. Lenstra. (1992). "A Local Search Template." In R. Manner and B. Manderick (eds.), *Parallel Problem Solving From Nature*. Belgique, pp. 67–76.

Voigt, H.-M., J. Born, and I. Santibanez-Koref. (1990). "Modelling and Simulation of Distributed Evolutionary Search Processes for Function Optimization." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature*. Vol. 496 of *LNCS*, Dortmund, Germany, Springer-Verlag, pp. 373–380.

Von Laszewski, G. and H. Muhlenbein. (1990). "Partitioning a Graph with Parallel Genetic Algorithm." In H.-P. Schwefel and R. Manner (eds.), *Parallel Problem Solving from Nature.* Vol. 496 of *LNCS*, Dortmund, Germany. Springer-Verlag, pp. 165–169.

Voss, S. (1993). "Tabu Search: Applications and Prospects." In *Network Optimization Problems*, World Scientific, USA, pp. 333–353.

Wang, L.-H., C.-Y. Kao, M. Ouh-Young, and W.-C. Chen, (1995). "Molecular Binding: A Case Study of the Population-Based annealing Genetic Algorithms." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 50–55.

Yang, S.Y., L.-J. Park, C.H. Park, and J.W. Ra. (1995). "A Hybrid Algorithm Using Genetic Algorithm and Gradient-Based Algorithm for Iterative Microwave Inverse Scattering." In *IEEE Int. Conf. on Evolutionary Computation ICEC'95*, Perth, Australia, pp. 450–455.