European Journal of Operational Research 199 (2009) 620–629

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Hybridizing exact methods and metaheuristics: A taxonomy

L. Jourdan *, M. Basseur, E.-G. Talbi

LIFL/INRIA/CNRS, Bat M3, Cité Scientifique, 59655 Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 22 September 2006
Accepted 10 July 2007
Available online 13 April 2008

Keywords:
Taxonomy
Combinatorial optimisation
Metaheuristics
Exact methods

ABSTRACT

The interest about hybrid optimization methods has grown for the last few years. Indeed, more and more papers about cooperation between heuristics and exact techniques are published. In this paper, we propose to extend an existing taxonomy for hybrid methods involving heuristic approaches in order to consider cooperative schemes between exact methods and metaheuristics. First, we propose some natural approaches for the different schemes of cooperation encountered, and we analyse, for each model, some examples taken from the literature. Then we recall and complement the proposed grammar and provide an annotated bibliography.

1. Introduction

NP-hard problems are difficult to solve and no polynomial time algorithm are known for solving them. Unfortunately, most combinatorial optimization problems, such as the Travelling Salesman, N-Queens, Bin Packing, 0/1 Knapsack, Graph Partitioning, are NP-hard. Two approaches can be considered to solve this kind of problems depending on their size.

For small instances, researchers usually use exact methods. Exact methods find the optimal solution and assess its optimality. There exist numerous exact methods such as the family of Branch and X (Branch and Bound algorithm [58], Branch and Cut algorithm [42], Branch and Price algorithm [12]), Linear Programming, Dynamic Programing, etc. A branch and X algorithm uses a divide and conquer strategy to partition the solution space into subproblems and then optimizes individually each subproblem. Exact methods are known to be time expensive, so they can not be applied to large NP-hard problems or difficult ones.

When instances become too large for exact methods, heuristics and in particular metaheuristics are often used. There are two main categories of metaheuristics: single solution algorithms and population based algorithms. The first category gathers local search (LS) [54], greedy heuristic (GH) [70], simulated annealing (SA) [50], tabu search (TS) [40], Iterated Local Search (ILS) [56] etc. The second category, which is more and more studied, regroups evolutionary algorithms such as genetic algorithms [44], evolution strategies [74], genetic programming [52], and also ant colonies (AC) [31], scatter search (SS) [39], immune systems [48] etc. However, in general, metaheuristics are not able to solve the problems to optimality and some convergence problems can be encountered.

During the last years, many works have been realized on cooperative (or hybrid) optimization approaches. In many cases, best results are obtained with this kind of approaches, especially on real-life problems. At the beginning, cooperations were mainly realized between several metaheuristics. But nowadays, more and more cooperation schemes between metaheuristics and exact approaches are proposed. These strategies usually give good results because they are able to exploit simultaneously the advantages of both types of methods. For example, it may allow to give quality guarantees to the identified solutions.

In this article, we propose to survey the different cooperation between these two types of method. The fact that more and more papers deal with this kind of approaches (see Fig. 1) clearly indicates that it is an important issue for the operational research community. So it seems interesting to classify these works. A state of the art of this type of cooperation has been proposed recently by Stützle and Dumitrescu [34]. They distinguish five classes of approaches for cooperation between exact and local search methods; they also provide an example for each type. The five classes proposed are:

- Use exact algorithms to explore large neighborhoods in local search algorithms.
- Perform several runs of a local search and exploit information in high quality solutions to define smaller problems that are amenable for solution with exact algorithms.
- Exploit bounds in constructive heuristics.
- Use information from relaxations of integer programing problems to guide local search or constructive algorithms.
- Use exact algorithms for specific procedures in hybrid metaheuristics.

* Corresponding author.
E-mail addresses: laetitia.jourdan@inria.fr (L. Jourdan), basseur@lifl.fr (M. Basseur), talbi@lifl.fr (E.-G. Talbi).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.07.035

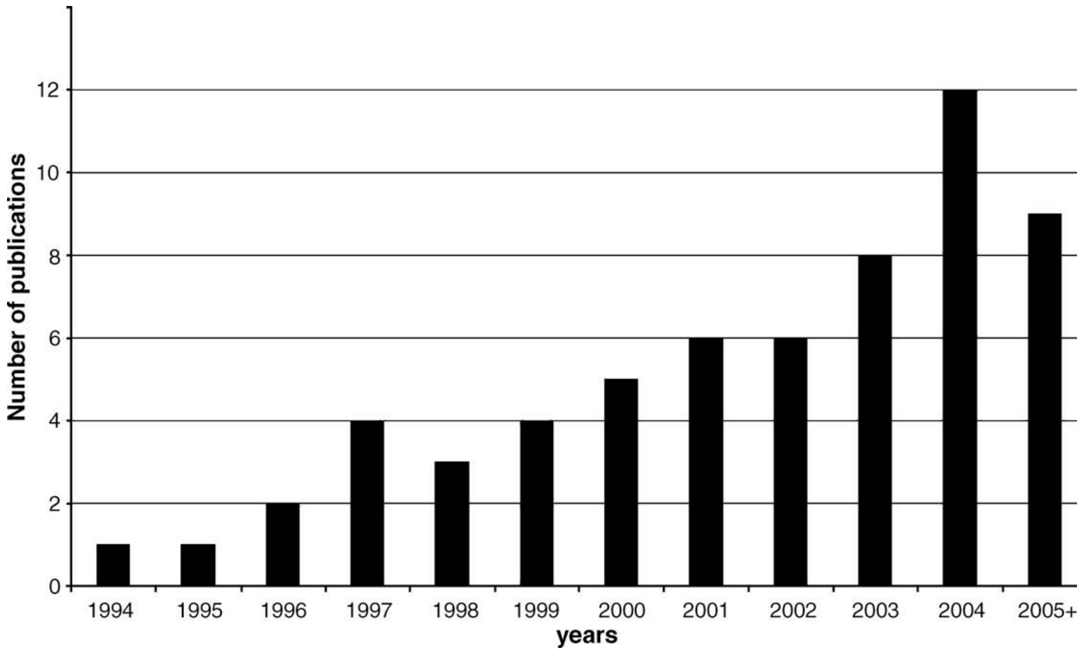

Fig. 1. The evolution of the publication activity on hybridizaton between exact methods and metaheuristics.

The survey proposed by Stützle and Dumitrescu presents several cooperative approaches to explain the different classes but would have more interest if it were generalized to every optimization methods. Their paper also excludes some combination such as preprocessing.

In [68], Puchinger and Raidl, propose a survey of the state-of-the-art approaches that combine exact methods and metaheuristics. Their survey provides a classification of methods thanks to different classes. The first one deals with collaborative combinations where no algorithm is contained in any other. This class is divided into subclasses:

- Sequential execution.
- Parallel and intertwined execution.

The second class regroups integrative combinations and is subdivided into two subclasses:

- Incorporating exact algorithms in metaheuristics.
- Incorporating metaheuristics in exact algorithms.

Puchinger and Raidl illustrate each subclass with examples issued from the literature.

In this article, we propose also to classify different articles issued from the literature but we will also propose a taxonomy of methods that combines exact and heuristic approaches. Important contributions of this article are the formal grammar proposed to classify the methods, integration of conceptual and hierarchical aspects. A separation between design and implementation is also taken into account. Our survey is far from providing an exhaustive list but it will constitute a good way for authors of new cooperation papers to classify their approach or for developers to find ideas on how to combine methods efficiently. In this article, cooperation and hybridization will be used in the same way. These terms will indicate algorithms which combine different optimization methods. The present paper uses the taxonomy proposed by Talbi [80] as we observe that it is valuable for cooperative methods between metaheuristics and exact approaches.

The remainder of the article is organized as follows. In Sections 2–4 the taxonomy used in [80] is recalled and illustrated with examples of cooperation between exact methods and metaheuristics, as in [80] the author only considers cooperation between metaheuristics. The taxonomy is divided into three general aspects: cooperation method design (Section 2), approach design (Section 3), and implementation issues (Section 4). In Section 5, the grammar for hybrid metaheuristics is extended, and an annotated review of different references is presented according to the taxonomy. Conclusions are drawn in Section 6.

2. Cooperation method design

Cooperation involves two main components: the design and the implementation. The former category concerns the cooperative algorithm itself, involving issues such as the functionality and the architecture. The implementation takes into account the hardware platform, programming model and the environment.

In this section, we will focus on the design of the cooperative mecanisms, i.e. how the methods will cooperate. For each type of classification, the derived classes are presented and some examples from the literature are described.

To facilitate the reading of our survey, the terms used in [80] are recalled. The design of metaheuristics can be classified in two types of design classification:

- Low-level/High-level
  - *Low-level*: The functional composition of a single optimization method. A given function of a metaheuristic is replaced by another method.
  - *High-level*: The different algorithms are self-contained.
- Relay/Teamwork
  - *Relay*: A set of methods is applied one after another, each using the output of the previous as its inputs, acting in a pipeline fashion.
  - Teamwork represents cooperative optimization models.

Four classes are derived from this hierarchical taxonomy (see Fig. 2).

2.1. LRH (Low-level Relay Hybrid)

This class corresponds to the algorithms in which a given method is embedded into another method; the embedded method has to be executed sequentially, i.e. the global method execution is dependant to the results obtained by the embedded method. This type of cooperation is commom when a heuristic approach is used to improve an exact approach. In the context of cooperation between metaheuristics, the most proposed approaches is to run an evolutionary algorithm then launch a local search in order to intensify the search on the best solutions. If we consider the cooperation between exact and heuristics methods, the most natural approach is to design an heuristic to improve the search strategy of the exact method (Fig. 3). For this scheme of cooperation, the heuristics will work on a problem which is from a different nature (node selection, column generation) that the considered optimization problem.

An example of this type of cooperation has been proposed by Augerat et al. [2]. In this study, a branch and cut algorithm is proposed to solve a capacitated vehicle routing problem (CVRP). The cutting plane generation is a crucial part of branch and cut algorithms. Indeed, it greatly determines their efficiency. The authors remark that the linear inequality resulting from the constraint capacities are those which provide the best cutting planes. So they propose different heuristic approaches (constructive heuristics, greedy algorithms, and Tabu search algorithms) to extract a set of violated capacity constraints of the relaxed problem.

This class of cooperation is not widely used for cooperative models between exact algorithms and heuristic approaches. Indeed, in many studies, the authors use simple (node exploration) or specific heuristics (column generation) to optimize the exact search strategy.

2.2. LTH (Low-level Teamwork Hybrid)

In this class, an element of a given method is replaced by another method. This kind of cooperation can drastically improve a metaheuristic. Oppositely to LRH cooperations, LTH consist in an embedded method which can be executed in parallel with the global method.

In the context of cooperation between metaheuristics, a well known LRH cooperative class of algorithms are memetic algorithm, i.e. genetic algorithms with a local search replacing a transformation operator, which is in many cases the mutation operator. Memetic algorithms are classified as LTH cooperation since the GA can be executed while a local search is running, applied on previously selected individuals from the GA population. Concerning the meta/exact cooperations, let us consider two main types of approaches:

- Exact search LTH cooperation (Fig. 4): the exact approach build partial solutions, which are used to define a search space for the heuristic approach. Then, the results obtained by the heuristic are analized in order to refine bounds, or column to generate in a branch & cut algorithm.
- Heuristic search LTH cooperation (Fig. 5): the heuristic search works like memetic algorithms, but in this case, the genetic operator is replaced by an exact search within a subspace of the global search space.

Cotta et al., propose a framework which lays on the cooperation between genetic algorithms and a Branch and Bound (B&B)

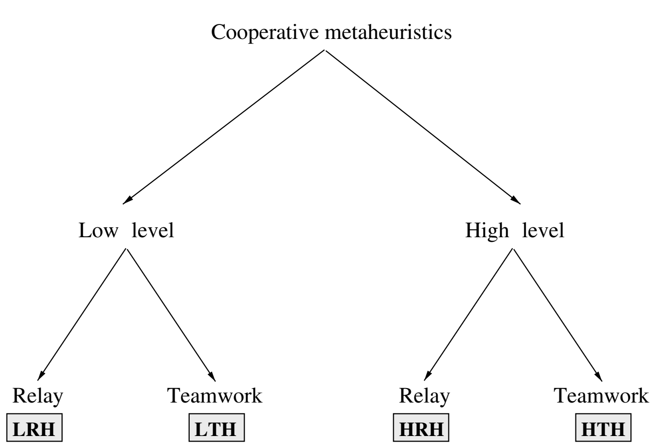

Fig. 2. The four classes derived from the cooperation method design classification: LRH, LTH, HRH, and HTH.

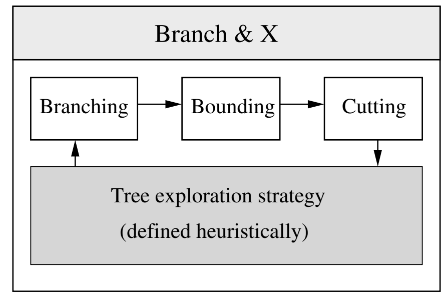

Fig. 3. LRH cooperation (exact search): Heuristic designed to define the exact search strategy (selection of the node to explore, column generation selection...). The heuristic is usually designed to help to define the exploration search strategy. (in gray).

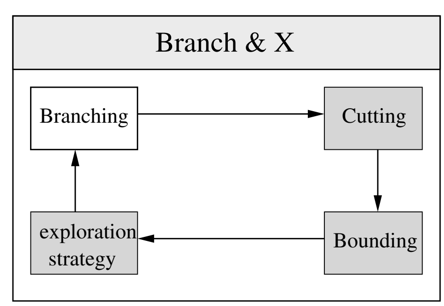

Fig. 4. LTH cooperation (exact search): Heuristic designed to explore the search space associated with a partial solution, in order to define bounds for cutting, or exploration strategy.

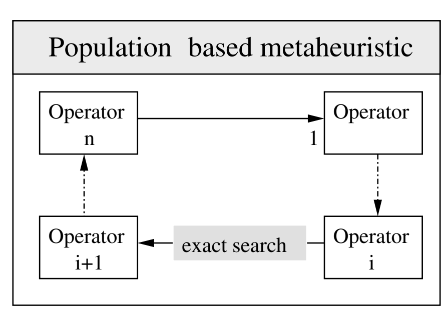

Fig. 5. LTH cooperation (heuristic search): Exact search is realized on solutions in order to intensify the search within an evolution search, with proof of the local optimality of the new solutions. The exact search could replace a genetic operator, like in memetic searches.

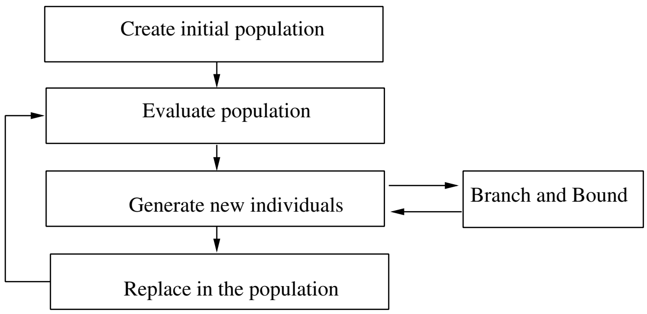

Fig. 6. Outline of the hybrid genetic algorithm of Cotta et al.: An example of Low-level TeamWork Hybrid (LTH).

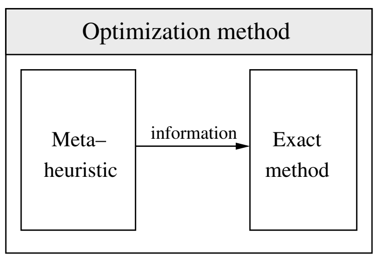

Fig. 7. HRH cooperation: some information is provided by the heuristic to the exact method. Two cases: (1) In exact search, some initial bounds/solutions are given. (2) In heuristics search, initial solutions are given to the exach algorithm which works on a subspace (partitioning, large neighborhood search).

algorithm which is used as an operator [26]. The resulting hybrid operator cleverly explores the dynastic potential (possible children) of the solutions being recombined, providing the best combination of formae (see Fig. 6).

In [49], Kostikas and Fragakis present the application of Genetic Programming (GP) in B&B based on Mixed Integer linear Programming (MIP). The hybrid architecture employs the GP as a node selection expression generator: a GP run, embedded into the B&B process, exploits the characteristics of the particular MIP problem being solved. The evolved method replaces the default one for the rest of the B&B.

Jahuira et al., propose different hybridizations between genetic algorithms and exact methods applied to the Travelling Salesman Problem (TSP) [45,46]. The cooperation is introduced in the genetic functions as the authors replace the genetic crossover by a branch and bound algorithm and a Minimal Spanning Tree solution algorithm. The initial solutions are also generated by means of the minimum spanning tree construction algorithm.

Large neighborhood search algorithms are typically *LTH* cooperations. These algorithms can be viewed as local search algorithms which use a large neighborhood to improve the efficiency of the search. The exploration of this large neighborhood can be either heuristic or exact. A survey of these methods can be found in [4]. Several studies propose exact methods to explore these large neighborhoods to find the best solution in a subspace of the global search space of the optimized problem. These types of approaches have been proposed by Bent and Van Hentenryck to solve the asymmetric TSP problem [16], or Shaw for VRP problem [78].

Globally, the frontier between LRH and LTH is thin. The difference mainly depends on the possibility, or not, to propose a parallel version of the proposed algorithm without applying drastical changes to the initial algorithm.

2.3. HRH (High-level Relay Hybrid)

In this class, the different methods are self-contained and are executed in sequence. This cooperation scheme is the most represented for general hybridization.

Of course, like in the other cooperation schemes, different types of resolution could be considered. However, in general, the most natural approach is to design a sequential execution of a metaheuristic which is launched before an exact approach (Fig. 7). The metaheuristic is designed in order to give information to the exact algorithm. If we consider an exact search, the information given could be initial bounds, for example, which helps the exact algorithm to speed up the search. If we consider a heuristic search, the metaheuristic gives initial solution(s) to the exact search, which helps to define a reduced search space to launch the exact search. For example, the search space could be reduced by defining partitions of the proposed solutions, or by defining large neighborhoods around the proposed solutions.

Klepeis et al., propose a cooperation between the alpha Branch and Bound algorithm and a conformational space annealing (CSA) algorithm for protein structure prediction [53]. The alpha branch and bound algorithm is a global optimization algorithm based on a branch and bound algorithm. It is applicable to a large class of nonlinear optimization problems that have twice differentiable functions [5]. The CSA is a stochastic method that employs elements of both simulated annealing and genetic algorithms [57].

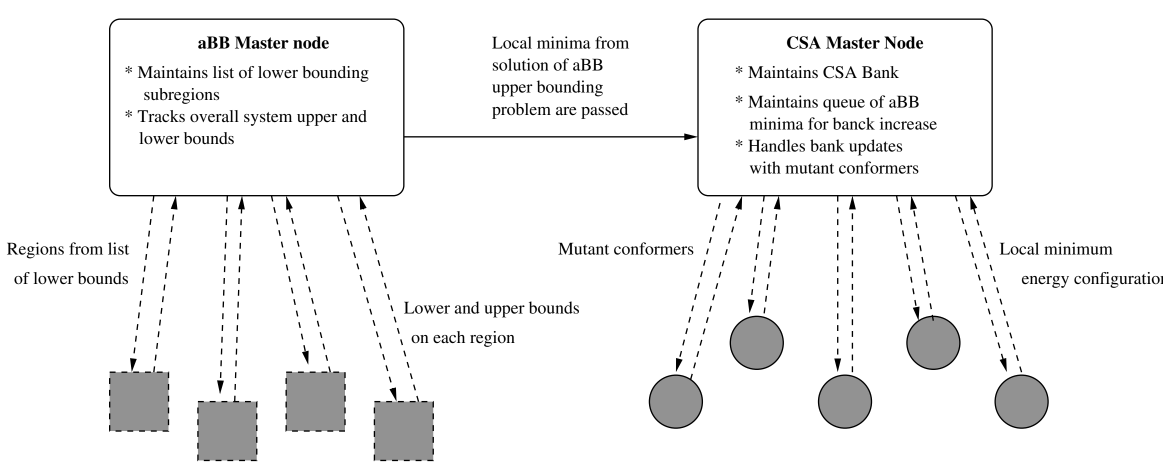

Fig. 8. An example of High-level Relay Hybrid for the protein structure prediction.

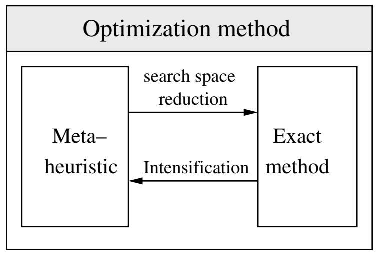

Fig. 9. HTH cooperation: two algorithms are launched in parallel, and exchange information (the metaheuristic provides information for search space reduction and the exact method provides improved solution to intensify the search).

In this algorithm, the authors alternate several runs of B& B and CSA. They parallelize their algorithm in a master–slave model (see Fig. 8).

In [16], Bent and Van Hentenryck propose a two-stage hybrid algorithm for the vehicle routing problem with time windows. First, the algorithm minimizes the number of vehicles by means of a simulated annealing algorithm. Then it minimizes the travel cost using a large neighborhood search technique which may relocate a large number of customers.

In [14], Basseur et al., investigate several cooperative approaches for a biobjective permutation flow-shop scheduling problem. These schemes are designed around a biobjective hybrid metaheuristic (Adaptive Genetic/Memetic Algorithm) and the two-phase method [84], a biobjective exact method. The main approach consists in fixing a part of initial solutions and optimizing exactly between two points of each individual. Then the process is iterated for each part of the best initial solutions proposed by the metaheuristic. Pareto fronts obtained by the Adaptive Genetic/Memetic Algorithm are strictly improved by this hybridization.

2.4. HTH (High-level Teamwork Hybrid)

This class contains algorithms where self-contained methods are performing a search in a parallel and cooperative manner.

Considering cooperation between metaheuristics, this cooperation involve principally island parallel models. It is almost the same with exact methods, with two different types of islands, those which are dedicated to exact search, and those dedicated to heuristic search. During the execution, the different algorithms exchange information, which is dependant of the type of the island (Fig. 9). The major difficulty is to set parameters (when and how the exchange is realized for example).

For example, in [62], Simulated Annealing and Branch and Bound are hybridized such that the two optimisation methods work on the same problem. The model exchanges information when conditions are satisfied. The model is parallel and is detailed in part 4.2.

In some cases, the two different cooperative approaches are not dedicated to solve instances of the same size. So, to obtain a *HTH* cooperation, the two approaches have often to solve different parts of the same problem in an independent manner. In [20], Chabrier et al. propose a cooperation between a local search and a column generation algorithm to solve VRP. The scheme of this *HTH* cooperation is shown in Fig. 10. The problem is divided into subproblems (partial model).

3. Approach design

Three criteria have been selected for the flat classification of cooperation between exact and heuristic methods: the complete resolution (exact or approximated), the resolution space (global or partial), and the nature of the cooperation (general or specialist).

3.1. Exact/approximated resolution

The type of the whole cooperative method could be either exact or heuristic. The exact cooperative approaches take useful information from heuristics to speed up the enumeration of the whole search space by upgrading bounds, finding initial solutions, defining useful cutting planes and so on.

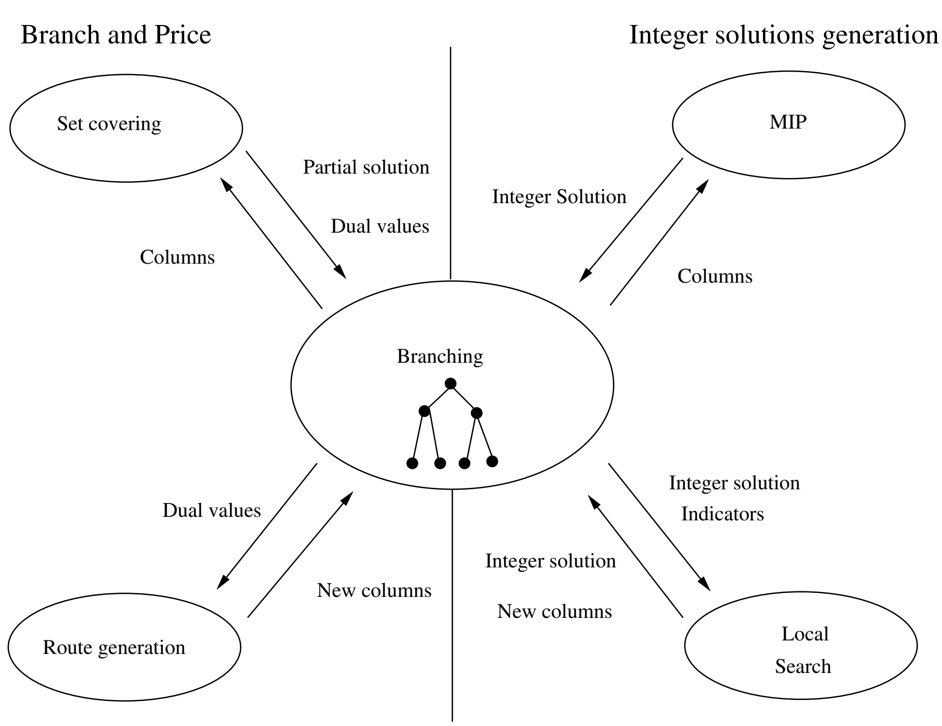

Fig. 10. An example of High-level Teamwork Hybrid scheme for the vehicle routing problem.

In the article of Chabrier et al. [20], a local search is used to generate new columns for a branch and cut algorithm. So the whole approach is exact, despite the use of a metaheuristic in the cooperative algorithm. Another way for this type of cooperation is the approach proposed by Cotta et al. [17]. The metaheuristic works on the same problem than the exact method and gives bounds to speed up the exact resolution.

Burke et al., describe a cooperative heuristic [8]. In this study, the exact approach is included in a local search mechanism to explore exactly interesting subspaces.

3.2. Global/partial cooperation

The components of the cooperative approach may work on the whole search space or only on a part. On one hand, in the global cooperation all the algorithms explore the same search space and on the other hand, in the partial cooperation, the problem is decomposed into sub-problems, each having its own search space. It may be noted that when the cooperation is a global one, the purpose is generally to solve the problem optimally, while partial cooperations are usually associated to approached solutions.

3.2.1. Partial hybrids

The majority of metaheuristic/exact cooperation approaches are often partial since the search space is generally too large for the exact method. Indeed, there are many examples of partial cooperation.

An example is the Mimausa method designed by Mautor and Michelon [60] for the quadratic assignment problem. The method builds at each iteration a subproblem and solves it by a branch and bound algorithm. Palpant et al., propose an heuristic for solving the Project scheduling with resource constraint. The heuristic integrates an exact method into a local search heuristic [65]. At each iteration, the process generates a main subproblem whose exact resolution provides a starting point for solving the remaining subproblem. Other approaches propose to solve the subproblems with an exact method and to integrate it in a metaheuristic: Budenbender for the Direct Flight Network Design Problem [11], the forget-and-extend algorithm [23] etc.

In [25], Chelouah and Siarry propose an hybrid method combining a tabu search algorithm and a simplex algorithm to deal with the global minimization of functions depending on continuous variables. The simplex algorithm is used to accelerate the convergence toward a minimum value.

In [14], branch and bound iterations are made on the potentially optimal Pareto solutions found by an hybrid evolutionary algorithm for a biobjective flowshop problem. However, only a small part of the search space is explored by the exact approach, because a large part of each individual is fixed before.

3.2.2. Global cooperation

In global cooperation, two optimization methods have to work on the same search space, usually the whole search space.

First, we consider as global cooperation exact algorithms which use a heuristic to improve bounds and then speed up an enumerative approach. In [72], Portmann et al., propose an exact approach to solve a hybrid flowshop problem. A specific heuristic computes solutions to provide an initial bound for a branch and bound algorithm. The two algorithms explore the same search space making this cooperation global.

3.3. Specialist/general

In a general hybrid model, all the algorithms solve the same optimization problem whereas the components of a specialist cooperation solve different problems. Many cooperations proposed in the literature are general, as for example the study of Cotta et al. [17].

The study of Chabrier et al. [20] is a good example of a specialist cooperation on the VRPTW (vehicle routing problem with time windows). One method solves a set covering problem and the other one a routing problem. In [83], T'kindt et al., investigate a biobjective flowshop problem with two machines. The objective are treated in a lexicographic way. The first objective, the *Makespan* is exactly solved by an exact approach (Johnson algorithm), then an ant system is applied to optimize the second objective. In many cases, specialist cooperations are used to solve problems with specific features, which can be solved exactly. Another possibility is a *LRH* approach, which optimizes heuristically (resp. exactly) a specific mechanism of an exact (resp. heuristic) method.

4. Implementation issues

The implementation choices of a cooperative algorithm can be a good way to improve the scalability of a method. Hence, an interesting point is to classify algorithms through the kind of implementation. In an optimization point of view, it is interesting to investigate the use of parallelism.

4.1. Sequential implementation

The majority of the proposed implementations are sequential. In many papers, it is suggested to parallelize the proposed cooperation in order to design better cooperation schemes and to speed up the execution. However, in these papers, the proposed approaches are not implemented, but only proposed as research perspectives. It is mainly due to the number of areas which are involved (parallelism, exact approaches, heuristics approaches...), and by the fact that the parallel models are not really natural (not like island models for genetic algorithms, for example).

4.2. Parallel implementation

Only a few studies propose parallel implementation for cooperation between exact and heuristics approaches. One very important point to determine is what kind of information is exchanged and when. In parallel implementations several architectures can be encountered: Multiprocessor, cluster or grid architecture. In metaheuristic and exact method cooperation the majority of the work use small cluster implementation. In [17], Cotta et al., propose a parallel implementation of their hybrid GA/B&B approach. GAs are executed in a parallel way, and a processor is dedicated to launch B&B resolutions on restricted problems. This processor allows to give new individuals to the GAs.

In [62], Simulated Annealing and Branch and Bound are hybridized in a parallel manner. The model uses two processors. On the first one, the simulated annealing is running and on the second one the Branch and Bound. Bounds and variable decisions are exchanged. Upper bounds obtained from SA are immediately passed onto the B&B code. If these bounds are better than B&B incumbent bounds, the B&B execution updates its current best bound value. Moreover, any integer bound obtained by the B&B execution is passed on to the SA code and used as an alternative reheated solution. Secondly, variable choice information is exchanged, once both SA and B&B have calculated their respective variable ranking and pseudocosts information, a single variable choice list is constructed by averaging the ranking positions from both strategies.

For future direction, it is important to notice that more and more frameworks allow to facilitate the parallelization of cooperation. To this purpose, we present several frameworks that allow to realize cooperation between metaheuristics and exact methods.

4.3. Frameworks

4.3.1. A-Teams

An asynchronous team (A-Team) is a strongly cyclic computational network [81]. The number of agents can be arbitrarily large and the agents may be distributed over an arbitrarily wide area. Agents cooperate by working on one another's results. Each agent is completely autonomous (it decides which results it is going to work on and when).

4.3.2. TECHS

TECHS (TEams for Cooperative Heterogeneous Search) allow the use of very different search agents within a search team while still achieving synergetic effects [32]. An agent could be either a metaheuristic or an exact method.

4.3.3. MALLBA and TRACER

n MALLBA (university of MAlaga, university of La Laguna, university of BArcelona), each optimization method is encapsulated in a skeleton [1]. Different exact methods are proposed (Branch and Bound, Divide and Conquer, Dynamic Programming) and also heuristic methods. MALLBA is one of the few frameworks that proposes the possibility to directly develop parallel hybrid methods between exact and heuristic methods. The parallelism is available for LAN and WAN computer platforms. TRACER is the following project of MALLBA.

Since many cooperation scheme could be designed to cooperate exact methods with metaheuristics, frameworks are useful in order to reuse mechanisms proposed by the literature. However, there exists only first experiments in this area since this research area is very recent.

5. Global overview

As we have extended the taxonomy [80] to the case of cooperation between metaheuristics and exact methods, we recall the grammar for cooperation schemes. In our case, we consider only heterogeneous models, since metaheuristics and exact methods are definitely different!

For the heuristics methods, we use these following abbreviations:

- SA: *Simulated Annealing*
- GA: *Genetic Algorithm*
- MA: *Memetic Algorithm*
- ES: *Evolution Strategy*
- GP: *Genetic Programming*
- NN: *Neural Network*
- DW: *Descent Walk*
- LS: *Local Search*
- ILS: *Iterated Local Search*
- TS: *Tabu Search*

<cooperative-algorithm >	⟶	*<design-issues><implementation-issue>*
<design-issues>	⟶	*<hierarchical><flat>*
<hierarchical>	⟶	*<LRH>* | *<LTH>* |
		<HRH> | *<HTH>*
<LRH>	⟶	LRH (*<metaheuristic>(<exact>))*|
		(*<exact>(<metaheuristic>))*
<HRH>	⟶	HRH (*<metaheuristic>+ <exact>*|
		<exact>+ <metaheuristic>)
<LTH>	⟶	LTH (*<metaheuristic> (<exact>))*|
		(*<exact>(<metaheuristic>))*
<HTH>	⟶	HTH (*<metaheuristic>+ <exact>)*|
		<exact>+ <metaheuristic>)
<flat>	⟶	(*<resolution>, <optimization>, <function>)*
<resolution>	⟶	Exact | Approximate
<optimization>	⟶	global | partial
<function>	⟶	general | specialist
<implementation-issue>	⟶	sequential | parallel | *<scheduling>*
<scheduling>	⟶	static | dynamic | adaptive
<metaheuristic>	⟶	SA | GA | MA | ES | GP | NN | DW | LS |
		ILS | TS | GH | AF | SS | CSA | SH | ... |
		<cooperative-algorithm>
<exact>	⟶	B&B | *alpha*B&B | B&C | B&P | ... |
		B&C&P | LP | MLP | DP | Splx | SG | SM |
		<cooperative-algorithm>

Fig. 11. The proposed grammar for cooperation between metaheuristics and exact methods.

- GH: *Greedy Heuristic*
- AC: *Ant Colonies*
- SS: *Scatter Search*
- CSA: *Conformational Space Annealing*
- SH: Specific Heuristic

For the exact methods, the abbreviations used are:

- B&B: Branch and Bound
- *alpha*B&B: the alpha Branch and Bound
- B&C: Branch and Cut
- B&C&P: Branch and Cut and Price
- BS: Beam Search (Breadth first search process without backtracking)
- LP: Linear Programming (as CPlex, etc.)
- DP: Dynamic Programming
- CP: Constraint Programming
- Splx: Simplex Search
- SG: Search Goal
- SES: Specific exact search

Table 1
Annotated bibliography

Refs.	Design	Optimization problem
[3]	LTH (DP(ILS)) (approximate, partial, general) sequential	TSP
[6]	HRH(B&P + SH) (approximate, partial, general) sequential	Large-scale set partitioning
[14]	HRH(HTH(GA + MA),B&B) (approximate, partial, general) sequential	Biobjective M-machines Flow-shop problem
[13]	HRH(HRH(GH + LS) + B&C) (exact, global, general) sequential	VRP with time windows
[10]	HRH(TS + LP) (approximate, partial, general) sequential	Irregular stock cutting
[16]	HRH(SA + LTH (LS(B&B))) (approximate, partial, general) sequential	VRP with time windows
[11]	HRH(GH + LTH (TS(B&B))) (approximate, partial, general) sequential	Direct flight network design
[8]	LRH(LS + DP) (approximate, partial, general) sequential	Asymmetric TSP
[17]	LTH(GA(B&B)) (approximate, partial, general) parallel static	TSP
[24]	LTH(LS(DP)) (approximate, partial, general) sequential	One machine Flow-shop (weighted sum of tardiness)
[19]	HRH(RS + LP) (approximate, partial, general) sequential	Fiber-optic cable manufacturing (scheduling)
[20]	HTH(B&P(HRH(MLP+LS))) (exact, global, specialist) sequential	VRP
[25]	HRH(TS + Splx) (approximate, partial, general) sequential	Multiminima continuous functions
[26]	LTH(GA(B&B)) (approximate, partial, general) sequential	Generalized Schwefel function, rule base learning
[28]	LTH(TS(B&B)) (approximate, partial, general) sequential	VRP
[33]	HTH(AS(HRH(B&B + AC))) (approximate, partial, specialist) sequential	Local access network design
[35]	HRH(LP + GA) (approximate, partial, general) sequential	Generalized assignment
[36]	LRH(B&B + GA) (exact, global, general) sequential	Max-SAT
[45]	LRH(MST + LTH (GA(B&B—ACM))) (approximate, global, general) sequential	TSP
[46]	LRH(MST + LTH (GA(ACM))) (approximate, global, general) sequential	TSP
[51]	HRH(LP + MA) (approximate, partial, general) sequential	Prize-collecting steiner tree
[53]	HRH(αB&B + CSA) (approximate, partial, specialist) parallel static	Protein structure prediction
[47]	LTH (GA(Splx)) (approximate, partial, general) sequential	Gene regulatory network models
[49]	LTH (MLP(GP)) (exact, global, specialist) sequential	MIPLIB3 benchmarks
[59]	LRH(AS + LP) (approximate, global, specialist) sequential	Quadratic assignment
[60]	HRH(B&B + TS) (approximate,partial,general) sequential	Quadratic assignment
[65]	HTH(LS(CP)) (approximate, partial,general) sequential	Project scheduling with resource constraints
[69]	HRH(DW + B&B) (exact, global,general) sequential	Quadratic assignment
[66]	HTH(LS(CP)) (approximate, partial,general) sequential	TSP
[72]	HRH(SH + (HTH(B&B(GA)))) (exact, global, general) sequential	Hybrid flow-shop
[76]	HRH(SH + B&B) (approximate, partial, general) sequential	*p-median* problem
[78]	LTH (LS(B&B)) (approximate, partial, general) sequential	VRP
[79]	HRH(B&B + NN) (approximate, partial, general) sequential	Biobjective broadcast scheduling problem
[83]	HRH(SES + GA) (approximate, partial, specialist) sequential	Biobjective 2 machines flow-shop problem
[85]	LRH(ILS + Splx) (approximate, global, specialist) sequential	One-dimensional cutting stock variant (pattern restricted problem)
[87]	HRH(AS + RT) (approximate, partial, general) sequential	0–1 multi-dimensional knapsack
[75]	LTH(LP(GA)) (exact, partial, general) sequential	graph coloring
[55]	HRH(GA + MIP) (exact, partial, general) sequential	Markov Decision Processes
[61]	LTH(GA(SES)) (approximate, partial, specialist) sequential	graph colouring
[63]	HRH(B&B + GA) (approximate, partial, specialist) sequential	Flowshop
[71]	HRH(IP + SS) (approximate, partial, specialist) sequential	0–1 programming
[67]	LTH(B&C&P(GA)) (approximate, partial, general) sequential	2D bin packing
[82]	HRH(GA + IP) (exact, partial, general) sequential	Jobshop
[37]	HTH(B&B + EA) approximate, partial, general sequential	Multidimensional Knapsack
[29]	HRH(IPL + AC) (approximate, partial, specialist) sequential	Project portfolio selection
[41]	LTH(SA(LP)) (approximate, partial, general) sequential	Irregular strip packing
[38]	LTH(SS(Splx)) (approximate, partial, general) sequential	Bicriteria 0, 1-knapsack
[27]	HTH(GA + LP) (approximate, partial, general) sequential	Scheduling in power systems
[73]	HTH (B&B + GA) (exact, global, general) sequential	Hybrid flowshops
[62]	HTH(SA + B&B) (approximate, global, general) parallel	Zero-one LP
[77]	HRH(Cplex + GA) (approximate, partial, general) sequential	Single line scheduling
[18]	HRH(LR + TS) (approximate, global, general) sequential	Single source capacitated location problem
[7]	HRH(BS + TS) (approximate, partial, general) sequential	One-dimensional bin packing problem
[9]	LTH(GRASP(OCTANE)) (approximate, partial, specialist) sequential	Multiconstraint knapsack problem
[21]	LTH(TS(LP)) (approximate, partial, specialist) sequential	Capacitated network design
[15]	LTH(AC(BS)) (approximate, global, general) sequential	open shop scheduling
[22]	LTH(LS(SES)) (approximate, partial, general) sequential	Routing
[30]	LTH(BS(LS)) (approximate, partial, general) sequential	2-machine flow shop, uncapacitated p-median location problem
[64]	HRH(LP + GRASP) (approximate, partial, specialist) sequential	Weighted maximal planar graph problem
[86]	LTH (LR + TS) (approximate, partial, general) sequential	Workshift and rest assignment of nursing personnel
[43]	LTH (LS(B&B)) (approximate, partial, general) sequential	Flow shop problem

The grammar is proposed in Fig. 11 and is particularly adapted to the design of cooperative schemes where it is interesting to know which method uses the other one. So the order in the description of a method has an importance. For example [17] is a Low Level Co-Evolutionary Hybrid scheme denoted by *LTH* (GA (B&B)) for the Design and this means that the B&B is used as an operator by the genetic algorithm. So for *LTH* scheme the first method uses the second in its design.

For the relay model, the order used indicates the order of the different methods used by the authors. For example in [14], the authors use in one of their models a *LRH*(GA + exact) which means that they apply an exact method on the solutions of a genetic algorithm in order to have the exact Pareto front in a multi-objective flowshop problem.

In Table 1, we propose a annoted survey of the literature, with for each reference the corresponding classification and grammar and the optimization problem solved.

6. Conclusion

The interest for hybrid metaheuristics is still growing as we can observe that several workshops and conferences are dealing with it: Hybrid Metaheuristics (HM 2005, 2006) workshop of ECAI, Conference on Hybrid Intelligent Systems (HIS 2005, 2006), Application of Hybrid Evolutionary Algorithms to NP-complete problems workshop of GECCO, etc.

In this paper, we have presented an extension of the work of Talbi [80] to cooperation between exact methods and metaheuristics. The objective was to offer a short review of the literature on the subject but also the paper had the purpose to show how the methods can be combined and how to describe such cooperations in order to favorize discussion in future papers on cooperation.

With this classification, we remark that several ways of cooperation can be explored. Moreover, only a few parallel implementation are realized for cooperative approaches between exact and heuristic approaches. Parallelism could be a good way to improve algorithms efficiency, as in many cases for hybrid metaheuristics [80]. A second important point is that most of the found cooperative algorithms involving on exact scheme are for mono-objective problems. The major difficulty for applying such cooperation to multi-objective problems is that there are few exact methods that can treat the specificities of multi-objective optimization [14].

In order to solve these difficulties, the use of frameworks seems to be a promizing way, if the user is not expert in all the topics involved in these cooperations. For example, some heuristics could be designed by an expert of this area, which also uses a framework in order to solve subproblems. The frameworks which could be involved concerns exact search, heuristics search, parallelism, or multi-objective optimization. It will be also very interesting to propose cooperative frameworks, which try to unify the different types of existing framework, in order to allow the user to define complex cooperative models in a few effort.

References

[1] E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Dfaz, I. Dorta, J. Abarr, J. Gonzlez, C. Len, L. Moreno, J. Petit, J. Roda, A. Rojas, F. Xhafa, Mallba: A library of skeletons for combinatorial optimisation, in: Proceedings of the Euro-Par, volume LNCS 2004, 2002, pp. 927–932.

[2] P. Augerat, J.M. Belenguer, E. Benavent, A. Corber'n, D. Naddef, Separating capacity constraints in the CVRP using tabu search, European Journal of Operational Research 106 (2–3) (1998) 546–557.

[3] D. Applegate, R. Bixby, V. Chvátal, W. Cook, Finding tours in the TSP. Technical Report 99885, Forschungsinstitut fnr Diskrete Mathematik, Universität Bonn, Germany, 1999.

[4] R.K. Ahuja, O. Ergun, J.B. Orlin, A.P. Punnen, A survey of very large-scale neighborhood search techniques, Discrete Applied Mathematics 123 (1-3) (2002) 75–102.

[5] I. Androulakis, C. Maranas, C. Floudas, ffbb: A global optimization method for general constrained nonconvex problems, Journal of Global Optimization 7 (1995) 337–363.

[6] A. Atamturk, G.L. Nemhauser, M.W.P. Savelsbergh, A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems, Journal of Heuristics 1 (1996) 247–259.

[7] A.C.F. Alvim, C. Ribeiro, F. Glover, D.J. Aloise, A hybrid improvement heuristic for the one-dimensional bin packing problem, Journal of Heuristics 10 (2) (2004) 205–229.

[8] E.K. Burke, P.I. Cowling, R. Keuthen, Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem, in: Applications of Evolutionary Computing. EvoWorkshops2001, Lecture Note in Computer Science, vol. 2037, Springer-Verlag, Como, Italy, 2001, pp. 203–212.

[9] M. Barake, P. Chardaire, G.P. McKeown, Metaheuristics: Computer decision-making, The Probe Metaheuristic and its Application to the Multiconstraint Knapsack Problem, Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[10] J.A. Bennell, K.A. Dowsland, Hybridising tabu search with optimisation techniques for irregular stock cutting, Management Science 47 (8) (2001) 1160–1172.

[11] K. Büdenbender, T. Grünert, H-J. Sebastian, A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem, Transportation Science 34 (4) (2000) 364–380.

[12] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: Column generation for solving huge integer programs, Operations Research (1998) 316–329.

[13] J.F. Bard, G. Kontoravdis, G. Yu, A branch-and-cut procedure for the vehicle routing problem with time windows, Transportation Science 36 (2002) 250–269.

[14] M. Basseur, J. Lemesre, C. Dhaenens, E-G. Talbi, Cooperation between branch and bound and evolutionary approaches to solve a biobjective flow shop problem, Workshop on Evolutionary Algorithms (WEA'04), vol. 3059, Springer-Verlag, 2004, pp. 72–86.

[15] C. Blum, Beam-aco – hybridizing ant colony optimization with beam search: An application to open shop scheduling, Computers and OR 32 (2005) 1565–1591.

[16] R. Bent, P. Van Hentenryck, A two-stage hybrid local search for the vehicle routing problem with time windows. Technical Report Tech Report CS-01-06, Brown University, USA, September 2001.

[17] C. Cotta, J.F. Aldana, A.J. Nebro, J.M. Troya, Hybridizing genetic algorithms with branch and bound techniques for the resolution of the tsp, in: D.W. Pearson, N.C. Steele, R.F. Albrecht (Eds.), Artificial Neural Nets and Genetic Algorithms 2, Springer-Verlag, 1995, pp. 277–280.

[18] M.J. Cortinhal, M.E. Captivo, Upper and lower bounds for the single source capacitated location problem, EJOR 151 (2) (2003) 333–351.

[19] D. Clements, J. Crawford, D. Joslin, G. Nemhauser, M. Puttlitz, M. Savelsbergh, Heuristic optimization: a hybrid AI/OR approach, in: Workshop on Industrial Constraint-Directed Scheduling, Schloss Hagenberg, Austria, 1997.

[20] A. Chabrier, E. Danna, C. Le Pape, Coopération entre génération de colonnes sans cycle et recherche locale appliquée au routage de véhicules, in: JNPC, 2002.

[21] T.G. Crainic, M. Gendreau, J.M. Farvolden, A simplex-based tabu search method for capacitated network design, INFORMS Journal on Computing 12 (3) (2000) 223–236.

[22] P.I. Cowling, Ralf Keuthen, Embedded local search approaches for routing optimization, Computers and OR 32 (2005) 465–490.

[23] Y. Caseau, F. Laburthe, Effective forget-and-extend heuristics for scheduling problems, in: CP-AI-OR'02 – Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 1999, Ferrara (Italy).

[24] R.K. Congram, C.N. Potts, S.L. van de Velde, An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem, INFORMS Journal on Computing 14 (1) (2002) 52–67.

[25] R. Chelouah, P. Siarry, A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multiminima functions, European Journal of Operational Research 161 (3) (2005) 636–654.

[26] C. Cotta, J.M. Troya, Embedding branch and bound within evolutionary algorithms, Applied Intelligence 18 (2003) 137–153.

[27] K.P. Dahal, C.J. Aldridge, S.J. Galloway, Evolutionary hybrid approaches for generation scheduling in power systems, European Journal of Operational Research (2006) [Available online, 25 January 2006].

[28] B. De Backer, V. Furnon, P. Kilby, P. Prosser, P. Shaw, Solving vehicle routing problems using constraint programming and metaheuristics, Journal of Heuristics 6 (4) (2000) 501–524.

[29] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection, European Journal of Operational Research 171 (3) (2006) 830–841.

[30] F. Della Croce, M. Ghirardi, R. Tadei, Recovering beam search: Enhancing the beam search approach for combinatorial optimization problems, Journal of Heuristics 10 (1) (2004) 89–104.

[31] M. Dorigo, V. Maniezzo, A. Colorni, Positive feedback as a search strategy. Technical Report No. 91016, Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie, 1991.

[32] J. Denzinger, T. Offermann, On cooperation between evolutionary algorithms and other search paradigms, in: CEC-99, 1999, pp. 2317–2324.

[33] C. Duhamel, A. Quilliot, Coupling a metaheuristic with an exact method for the local access network design problem, In Optimization 2001, 2001.
[34] I. Dumitrescu, T. Stützle, Combinations of local search and exact algorithms, in: EvoWorkshops 2003, 2003, pp. 211–223.
[35] H. Feltl, G.R. Raidl, An improved hybrid genetic algorithm for the generalized assignment problem, Symposium on Applied Computing, SAC'04, ACM Press, 2004, pp. 990–995.
[36] A.P. French, A.C. Robinson, J.M. Wilson, Using a hybrid genetic-algorithm/branch and bound approach to solve feasibility and optimization integer programming problems, Journal of Heuristics 7 (6) (2001) 551–564.
[37] J.E. Gallardo, C. Cotta, A.J. Fernández, Solving the multidimensional knapsack problem using an evolutionary algorithm hybridized with branch and bound, in: J. Mira, J.R. Álvarez (Eds.), IWINAC (2), Lecture Notes in Computer Science, vol. 3562, Springer, 2005, pp. 21–30.
[38] C. Gomes da Silva, J. Figueira, J. Climaco, Integrating partial optimization with scatter search for solving bi-criteria 0,1-knapsack problems, European Journal of Operational Research (2006) (Available online, 17 February 2006).
[39] F. Glover, Heuristics for integer programming using surrogate constraints, Decision Sciences 8 (1) (1977) 56–166.
[40] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 13 (1986) 533–549.
[41] A.M. Gomes, J.F. Oliveira, Solving irregular strip packing problems by hybridising simulated annealing and linear programming, European Journal of Operational Research 171 (3) (2006) 811–829.
[42] R. Gomory, Outline of an algorithm for integer solutions to linear programs, Bulletin of the American Mathematical Society 64 (1958) 275.
[43] M. Haouari, T. Ladhari, A branch-and-bound-based local search method for the flow shop problem, Journal of the Operational Research Society 54 (10) (2003) 1076–1084.
[44] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[45] C.A.R. Jahuira, Hybrid genetic algorithm with exact techniques applied to tsp, in: Second International Workshop on Intelligent Systems Design and Application, Dynamic Publishers, 2002, pp. 119–124.
[46] C.A.R. Jahuira, E. Cuadros-Vargas. Solving the tsp by mixing gas with minimal spanning tree, in: Ernesto Cuadros-Vargas, Eduardo Tejada-Gamero, Adenilso da Silva Simao, (Eds.), First International Conference of the Peruvian Computer Society, Lima – Perù, Peruvian Computer Society, January 2003, pp. 123–123.
[47] P. Koduru, S. Das, S. Welch, J.L. Roe, A multi-objective ga-simplex hybrid approach for gene regulatory network models, in: Congress on Evolutionary Computation (CEC'04), vol. 2, IEEE Service Center, Portland, Oregon, USA, June 2004, pp. 2084–2091.
[48] J.O. Kephart, A biologically inspired immune system for computers, in: Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Cambridge, MA, US, MIT Press, 1994, pp. 130–139.
[49] K. Kostikas, C. Fragakis, Genetic programming applied to mixed integer programming, in: EuroGP 2004, Lecture Notes in Computer Science, vol. 3003, 2004, pp. 113–124.
[50] S. Kirkpatrick, D.C. Gelatt, M.P. Vechhi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[51] G.W. Klau, I. Ljubic, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, R. Weiskircher, Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem, in: K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, P. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, A. Tyrrell, (Eds.), Lecture Note in Computer Science, Proceedings of the 2004 Genetic and Evolutionary Computation Conference, vol. 3102, Seattle, Washington, USA, June 2004, pp. 1304–1315.
[52] J.R. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge, MA, 1992.
[53] J.L. Klepeis, M.J. Pieja, C.A. Floudas, Hybrid global optimization algorithms for protein structure prediction: Alternating hybrids, Biophysical Journal 4 (84) (2003) 869–882.
[54] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Reinhart, and Winston, New York, 1976.
[55] Z-Z. Lin, J.C. Bean, C.C. White III, A hybrid genetic/optimization algorithm for finite-horizon, partially observed markov decision processes, INFORMS Journal on Computing 16 (1) (2004) 27–38.
[56] H.R. Lourenço, O. Martin, T. Stützle, Handbook of Metaheuristics, Iterated Local Search, Kluwer Academic Publishers, Norwell, MA, 2002.
[57] J. Lee, H. Scheraga, S. Rackovsky, New optimization method for conformational energy calculations on polypeptides: Conformational space annealing, Journal of Computational Chemistry 18 (1997) 1222–1232.
[58] E.L. Lawler, D.E. Wood, Branch and bound methods: A survey, Operations Research 14 (1966) 699–719.
[59] V. Maniezzo, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, INFORM Journal on Computing 11 (4) (1999) 358–369.
[60] T. Mautor, P. Michelon, Mimausa: A new hybrid method combining exact solution and local search, in: Second International Conference on Meta-Heuristics (MIC'97), 1997.
[61] A. Marino, A. Prugel-Bennett, C. Glass, Improving graph colouring with linear programming and genetic algorithms, in: K. Miettinen, M.M. Makela, J. Toivanen (Eds.), EUROGEN99, Jyvaskyla, Finland, 1999, pp. 113–118.
[62] V. Nwana, K. Darby-Dowman, G. Mitra, A co-operative parallel heuristic for mixed zero-one linear programming: Combining simulated annealing with branch and bound, European Journal of Operational Research 164 (2005) 12–23.
[63] A. Nagar, S.S. Heragu, J. Haddock, A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem, Annals of Operations Research 63 (1996) 397–414.
[64] I.H. Osman, M. Hasan, A. Abdullah, Lp-based meta-heuristics for the weighted maximal planar graph problem, Journal of Operational Research Society 53 (10) (2002) 1142–1149.
[65] M. Palpant, C. Artigues, P. Michelon, LSSPER: Solving the resource-constrained project scheduling problem with large neighbourhood search, Annals of Operations Research 31 (1) (2004) 237–257.
[66] G. Pesant, P. Gendreau, A constraint programming framework for local search methods, Journal of Heuristics 5 (3) (1999) 255–279.
[67] J. Puchinger, G.R. Raidl, An evolutionary algorithm for column generation in integer programming: An effective approach for 2d bin packing, In PPSN VIII, 2004, pp. 642–651.
[68] J. Puchinger, G.R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification, in: IWINAC, pp. 41–53, 2005.
[69] P.M. Pardalos, K.G. Ramakrishnan, M.G.C. Resende, Y. Li, Implementation of a variance reduction based lower bound in a branch and bound algorithm for the quadratic assignment problem, SIAM Journal on Optimization 7 (1997) 280–294.
[70] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, 1982.
[71] A. Plateau, D. Tachat, P. Tolla, A hybrid search combining interior point methods and metaheuristics for 0–1 programming, International Transactions in Operational Research 9 (6) (2002) 731–746.
[72] M.C. Portmann, A. Vignier, D. Dardihac, D. Dezalay, Branch and bound crossed with ga to solve hybrid flowshops, European Journal of Operational Research 107 (2) (1998) 389–400.
[73] M.C. Portmann, A. Vignier, D. Dardilhac, D. Dezalay, Branch and bound crossed with ga to solve hybrid flowshops, European Journal of Operational Research 107 (1998) 389–400.
[74] I. Rechenberg, Evolutions Strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973.
[75] G. Ribeiro Filho, L.A.N. Lorena, Constructive genetic algorithm and column generation: An application to graph coloring, in: APORS 2000 – The Fifth Conference of the Association of Asian-Pacific Operations Research Societies within IFORS, 2000.
[76] C. Rosing, C. ReVelle, Heuristic concentration: Two stage solution construction, European Journal of Operational Research 97 (1997) 75–86.
[77] R. Spina, L.M. Galantucci, M. Dassisti, A hybrid approach to the single line scheduling problem with multiple products and sequence-dependent time, Computers and Industrial Engineering 45 (4) (2003) 573–583.
[78] P. Shaw, Using constraint programming and local search methods to solve vehicle routing problems, in: Principles and Practice of Constraint Programming – CP98, 4th International Conference, Lecture Notes in Computer Science, vol. 1520, Springer-Verlag, Pisa, Italy, 1998, pp. 417–431.
[79] H. Shi, L. Wang, A mixed branch-and-bound and neural network approach for the broadcast scheduling problem, in: Design and Application of Hybrid Intelligent Systems, HIS'03, IOS Press, 2003, pp. 42–49.
[80] E-G. Talbi, A taxonomy of hybrid metaheuristics, Journal of Heuristics 8 (2) (2002) 541–564.
[81] S. Talukdar, L. Baerentzen, A. Gove, P.S. de Souza, Asynchronous teams: Cooperation schemes for autonomous agents, Journal of Heuristics 4 (4) (1998) 295–321.
[82] H. Tamura, A. Hirahara, I. Hatono, M. Umano, An approximate solution method for combinatorial optimization – a hybrid approach of genetic algorithm and Lagrange relaxation method, Transactions of the Society of Instrument and Control Engineers (Japan) 30 (3) (1994) 329–336.
[83] V. T'kindt, N. Monmarché, F. Tercinet, D. Laügt, An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem, European Journal of Operational Research 142 (2002) 250–257.
[84] E.L. Ulungu, J. Teghem, The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems, Foundations of Computing and Decision Sciences 20 (2) (1995) 49–165.
[85] S. Umetani, M. Yagiura, T. Ibaraki, One-dimensional cutting stock problem with a given number of setups: a hybrid approach of metaheuristics and linear programming, in: Hybrid Metaheuristics HM'04, at workshop European Conference on Artificial Intelligence ECAI'04, Valencia Spain, August 2004, pp. 101–114.
[86] C. Valouxis, E. Housos, Hybrid optimization techniques for the workshift and rest assignment of nursing personnel, Artificial Intelligence in Medicine 20 (2) (2000) 55–175.
[87] M. Vasquez, J.-K. Hao, A hybrid approach for the 0/1 multidimensional knapsack problem, in: IJCAI, 2001, pp. 328–333.