LNCS 4030

Francisco Almeida María J. Blesa Aguilera
Christian Blum José Marcos Moreno Vega
Melquíades Pérez Pérez Andrea Roli
Michael Sampels (Eds.)

Hybrid Metaheuristics

Third International Workshop, HM 2006
Gran Canaria, Spain, October 2006
Proceedings

Springer

Francisco Almeida   María J. Blesa Aguilera
Christian Blum   José Marcos Moreno Vega
Melquíades Pérez Pérez
Andrea Roli   Michael Sampels (Eds.)

Hybrid Metaheuristics

Third International Workshop, HM 2006
Gran Canaria, Spain, October 13-14, 2006
Proceedings

Springer

Volume Editors

Francisco Almeida
José Marcos Moreno Vega
Melquíades Pérez Pérez
DEIOC Universidad de La Laguna
Escuela Técnica Superior en Ingeniería Informática
Avda. Astrofísico Francisco Sánchez, s/n, 38271 La Laguna, Tenerife, Spain
E-mail: {falmeida, jmmoreno, melperez}@ull.es

María J. Blesa Aguilera
Christian Blum
Universitat Politècnica de Catalunya, ALBCOM research group
Omega Campus Nord, Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: {mjblesa, cblum}@lsi.upc.edu

Andrea Roli
Università degli Studi "G. D'Annunzio"
Dipartimento di Scienze
Viale Pindaro 42, 65127 Pescara, Italy
E-mail: a.roli@unich.it

Michael Sampels
Université Libre de Bruxelles
IRIDIA CP 194/6
Avenue Franklin D. Roosevelt 50, 1050 Bruxelles, Belgium
E-mail: msampels@ulb.ac.be

Library of Congress Control Number: 2006933415

CR Subject Classification (1998): F.2, F.1, G.1.6, G.1.2, G.2.1, I.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-46384-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-46384-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11890584 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4030

Preface

The International Workshop on Hybrid Metaheuristics reached its third edition with HM 2006. The active and successful participation in the past editions was a clear indication that the research community on metaheuristics and related areas felt the need for a forum to discuss specific aspects of hybridization of metaheuristics.

The selection of papers for HM 2006 consolidated some of the mainstream issues that have emerged from the past editions. Firstly, there are prominent examples of effective hybrid techniques whose design and implementation were motivated by challenging real-world applications. We believe this is particularly important for two reasons: on the one hand, researchers are conscious that the primary goal of developing algorithms is to solve relevant real-life problems; on the other hand, the path toward efficient solving methods for practical problems is a source of new outstanding ideas and theories.

A second important issue is that the research community on metaheuristics has become increasingly interested in and open to techniques and methods known from artificial intelligence (AI) and operations research (OR). So far, the most representative examples of such integration have been the use of AI/OR techniques as subordinates of metaheuristic methods. As a historical and etymological note, this is in perfect accordance with the original meaning of a *metaheuristic* as a "general strategy controlling a subordinate heuristic."

The awareness of the need for a sound experimental methodology is a third keypoint. This aspect has gained more relevance and currency, even though there are still no widely agreed standard methodologies. As research on hybrid metaheuristics is mostly based on experimental methods, similar standards to those found in the evaluation of experiments in natural sciences can be expected.

Scientific testing, a fourth notable aspect, emerges as a fundamental methodology for understanding the behavior of algorithms. The goal of scientific testing is to abstract from actual implementations and study, empirically and through predictive models, the effect of algorithmic components. This research approach can be particularly useful in the case of conjectures on metaheuristic algorithm behavior that, while being widespread in the community, have not yet been the subject of validation.

Finally, a tendency to reconsider hybrid metaheuristics from a higher and more general perspective is emerging. Providing classifications, systematic analyses and surveys on important branches underlines a certain maturity of the relatively young field.

This progression can be observed by an increasing number of submissions to the workshop: we received 42 paper submissions to HM 2006. Each submitted paper was sent to at least three reviewers. We are very grateful to the members of the Program Committee and the additional reviewers for the effort they made in carefully examining the papers and for the many valuable comments and suggestions they gave to the authors. Based on their comments, we finally accepted 13 submissions for publication and for presentation at HM 2006, resulting in an acceptance rate of roughly 31 %. In addition, we got one invited paper. The selection of papers was rather strict in order to guarantee the high quality of the proceedings and the workshop itself. We would like to thank all authors for their interest in our workshop.

The field of hybrid metaheuristics is the result of the composition of numerous streams in the field of algorithmics. However, these streams have increasingly come together and the main issues and characteristics of the field have evolved more clearly. For the future, we envision a scenario in which some challenges have to be faced:

- It should become common practice that experimental analysis meets high quality standards. This empirical approach is absolutely necessary to produce objective and reproducible results and to anchor the successes of metaheuristics in real-world applications.
- Hybrid metaheuristic techniques have to be openly compared not just among themselves but also with state-of-the-art methods, from whatever field they are. By following this approach, researchers would be able to design techniques that meet the goal of solving a real-world problem and to consider the other approaches as rich sources of design components and ideas.
- Scientific testing and theoretical models of algorithms for studying their behavior are still confined to a limited area of research. We believe that, by being able to explain rigorously algorithm behavior by means of sound empirical investigation and formal models, researchers would give the field a firmer status and give support to the development of real-world applications.

The achievement of these goals will take some time in view of the difficult theoretical and practical problems involved in these challenges. Nevertheless, research is very active and has already produced some remarkable results and studies in this direction.

August 2006

Francisco Almeida
María J. Blesa
Christian Blum
J. Marcos Moreno
Melquíades Pérez
Andrea Roli
Michael Sampels

Organization

Program Chairs

María J. Blesa	Universitat Politècnica de Catalunya, Barcelona, Spain
Christian Blum	Universitat Politècnica de Catalunya, Barcelona, Spain
Andrea Roli	Università degli Studi "G. D'Annunzio", Chieti-Pescara, Italy
Michael Sampels	Université Libre de Bruxelles, Belgium

Workshop Chairs and Local Organization

Francisco Almeida	Universidad de La Laguna, Tenerife, Spain
J. Marcos Moreno	Universidad de La Laguna, Tenerife, Spain
Melquíades Pérez	Universidad de La Laguna, Tenerife, Spain

Program Committee

Thomas Bartz-Beielstein	Universität Dortmund, Germany
Mauro Birattari	Université Libre de Bruxelles, Belgium
Ralf Bruns	Fachhochschule Hannover, Germany
Francisco Chicano	Universidad de Málaga, Spain
Óscar Cordón	Universidad de Granada, Spain
Carlos Cotta	Universidad de Málaga, Spain
Luca Di Gaspero	Università degli Studi di Udine, Italy
Marco Dorigo	Université Libre de Bruxelles, Belgium
Joshua Knowles	University of Manchester, UK
Andrea Lodi	Università degli Studi di Bologna, Italy
Vittorio Maniezzo	Università degli Studi di Bologna, Italy
Belén Melián Batista	Universidad de La Laguna, Spain
Daniel Merkle	Universität Leipzig, Germany
Bernd Meyer	Monash University, Australia
Martin Middendorf	Universität Leipzig, Germany
José A. Moreno	Universidad de La Laguna, Spain
David Pelta	Universidad de Granada, Spain
Steven Prestwich	4C, Cork, Ireland
Günther Raidl	Technische Universität Wien, Austria
Andrea Schaerf	Università degli Studi di Udine, Italy
Thomas Stützle	Technische Universität Darmstadt, Germany

El-Ghazali Talbi	École Polytechnique Universitaire de Lille, France
Fatos Xhafa	Universitat Politècnica de Catalunya, Spain
Pascal Van Hentenryck	Brown University, Providence, USA
José Luis Verdegay	Universidad de Granada, Spain

Additional Referees

Dan Ashlock, Emilie Danna, Marta Kasprzak, Michele Monaci, Alena Shmygelska, Peter J. Stuckey, Hande Yaman

Table of Contents

A Unified View on Hybrid Metaheuristics*

Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
raidl@ads.tuwien.ac.at

Abstract. Manifold possibilities of hybridizing individual metaheuristics with each other and/or with algorithms from other fields exist. A large number of publications documents the benefits and great success of such hybrids. This article overviews several popular hybridization approaches and classifies them based on various characteristics. In particular with respect to low-level hybrids of different metaheuristics, a unified view based on a common pool template is described. It helps in making similarities and different key components of existing metaheuristics explicit. We then consider these key components as a toolbox for building new, effective hybrid metaheuristics. This approach of thinking seems to be superior to sticking too strongly to the philosophies and historical backgrounds behind the different metaheuristic paradigms. Finally, particularly promising possibilities of combining metaheuristics with constraint programming and integer programming techniques are highlighted.

1 Introduction

Metaheuristics have proven to be highly useful for approximately solving difficult optimization problems in practice. A general overview on this research area can be found e.g. in [1], for more information see also [2,3]. The term *metaheuristic* was first introduced by Glover [4]. Today, it refers to a broad class of algorithmic concepts for optimization and problem solving, and the boundaries are somewhat fuzzy. Voß [5] gives the following definition:

> A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method.

According to Glover [2],

> ...these methods have over time also come to include any procedure for problem solving that employs a strategy for overcoming the trap of

* This work is supported by the European RTN ADONET under grant 504438.

F. Almeida et al. (Eds.): HM 2006, LNCS 4030, pp. 1–12, 2006.

local optimality in complex solution spaces, especially those procedures that utilize one or more neighborhood structures as a means of defining admissible moves to transition from one solution to another, or to build or destroy solutions in constructive and destructive processes.

Simulated annealing, tabu search, evolutionary algorithms like genetic algorithms and evolution strategies, ant colony optimization, estimation of distribution algorithms, scatter search, path relinking, the greedy randomized adaptive search procedure (GRASP), multi-start and iterated local search, guided local search, and variable neighborhood search are – among others – often listed as examples of classical metaheuristics, and they have individual historical backgrounds and follow different paradigms and philosophies; see e.g. [2].

Especially over the last years a large number of algorithms were reported that do not purely follow the concepts of one single traditional metaheuristic, but they combine various algorithmic ideas, sometimes also from outside of the traditional metaheuristics field. These approaches are commonly referred to as *hybrid metaheuristics*.

As for metaheuristics in general, there exist various perceptions of what a hybrid metaheuristic actually is. Looking up the meaning of *hybrid* in the current issue (May 2006) of the Merriam Webster dictionary yields

a) something heterogeneous in origin or composition,
b) something (as a power plant, vehicle, or electronic circuit) that has two different types of components performing essentially the same function,

while the current entry in Wiktionary defines this term as

a) offspring resulting from cross-breeding different entities, e.g. different species,
b) something of mixed origin or composition.

The motivation behind such hybridizations of different algorithmic concepts is usually to obtain better performing systems that exploit and unite advantages of the individual pure strategies, i.e. such hybrids are believed to benefit from synergy. The vastly increasing number of reported applications of hybrid metaheuristics and dedicated scientific events such as the series of *Workshops on Hybrid Metaheuristics* [6,7] document the popularity, success, and importance of this specific line of research. In fact, today it seems that choosing an adequate hybrid approach is determinant for achieving top performance in solving most difficult problems.

Actually, the idea of hybridizing metaheuristics is not new but dates back to the origins of metaheuristics themselves. At the beginning, however, such hybrids were not so popular since several relatively strongly separated and even competing communities of researchers existed who considered "their" favorite class of metaheuristics "generally best" and followed the specific philosophies in very dogmatic ways. For example, the evolutionary computation community grew up in relative isolation and followed relatively strictly the biologically oriented thinking. It is mostly due to the no free lunch theorems [8] that this situation fortunately changed and people recognized that there cannot exist a general optimization strategy which is globally better than any other. In fact, to solve a problem at hand most effectively, it almost always requires a specialized algorithm that needs to be compiled of adequate parts.

Several publications exist which give taxonomies for hybrid metaheuristics or particular subcategories [9,10,11,12,13,14]. The following section tries to merge the most important aspects of these classifications and at some points extends these views. Also, several examples of common hybridization strategies are given. In Section 3, we turn to a unified view on metaheuristics by discussing the pool template. It helps to extract the specific characteristics of the individual classical metaheuristics and to interpret them as a toolbox of key components that can be combined in flexible ways to build an effective composite system. Section 4 refers to a selection of highly promising possibilities for combining metaheuristics with algorithms from two other prominent research fields in combinatorial optimization, namely constraint programming and integer linear programming. Finally, conclusions are drawn in Section 5.

2 Classification of Hybrid Metaheuristics

Figure 1 illustrates the various classes and properties by which we want to categorize hybrids of metaheuristics. Hereby, we combine aspects from the taxonomy introduced by Talbi [10] with the points-of-view from Cotta [9] and Blum et al. [11]. Classifications with particular respect to parallel metaheuristics are partly adopted from El-Abd and Kamel [14] and Cotta et al. [12] and with respect to the hybridization of metaheuristics with exact optimization techniques from Puchinger and Raidl [13].

We start by distinguishing *what* we hybridize, i.e. which kind of algorithms. We might combine (a) different metaheuristic strategies, (b) metaheuristics with certain algorithms specific for the problem we are considering, such as special simulations, or (c) metaheuristics with other more general techniques coming from fields like operations research (OR) and artificial intelligence (AI). Prominent examples for optimization methods from other fields that have been successfully combined with metaheuristics are exact approaches like branch-and-bound, dynamic programming, and various specific integer linear programming techniques on one side and soft computation techniques like neural networks and fuzzy logic on the other side.

Beside this differentiation, previous taxonomies of hybrid metaheuristics [10,9] primarily distinguish the *level* (or strength) at which the different algorithms are combined: High-level combinations in principle retain the individual identities of the original algorithms and cooperate over a relatively well defined interface; there is no direct, strong relationship of the internal workings of the algorithms.

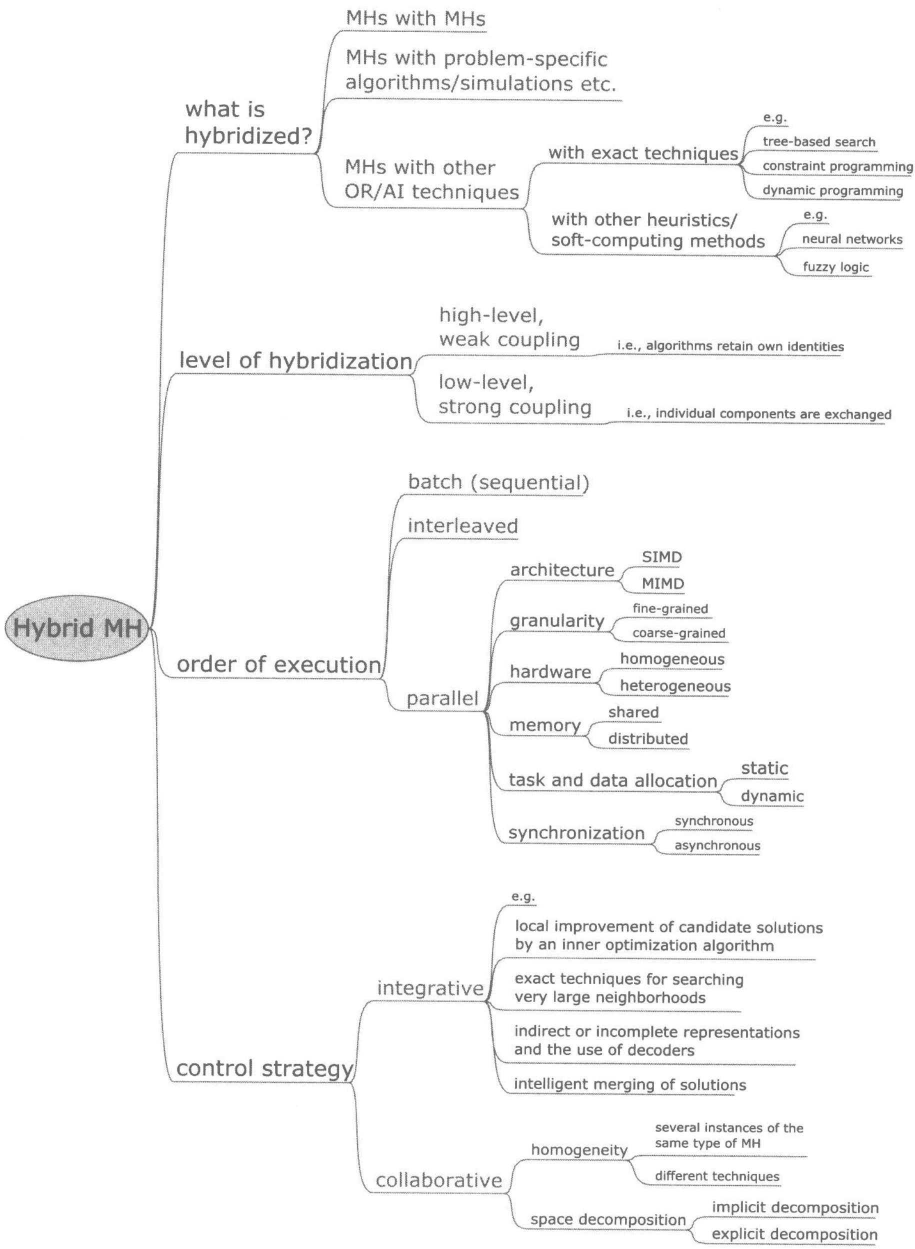

Fig. 1. A summarized classification of hybrid metaheuristics (MHs)

On the contrary, algorithms in low-level combinations strongly depend on each other – individual components or functions of the algorithms are exchanged.

Another property by which we may distinguish hybrid systems is the *order of execution*. In the batch model, one algorithm is strictly performed after the other, and information is passed only in one direction. An intelligent preprocessing of input data or a postprocessing of the results from another algorithm would fall into this category. Another example are multi-level problems which are solved by considering one level after the other by dedicated optimization algorithms. On the contrary, we have the interleaved and parallel models, in which the algorithms might interact in more sophisticated ways. Parallel metaheuristics are nowadays a large and important research field for their own, see [15]. Detailed classifications of hybrid parallel metaheuristics can be found in [14,12]. Following general characterizations of parallel algorithms, we can distinguish the architecture (SIMD: single instruction, multiple data streams versus MIMD: multiple instruction, multiple data streams), the granularity of parallelism (fine- versus coarse-grained), the hardware (homogeneous versus heterogeneous), the memory strategy (shared versus distributed memory), the task and data allocation strategy (static versus dynamic), and whether the different tasks are synchronized or run in an asynchronous way.

We can further distinguish hybrid metaheuristics according to their *control strategy*. Following [9,13], there exist integrative (coercive) and collaborative (cooperative) combinations.

In integrative approaches, one algorithm is considered a subordinate, embedded component of another algorithm. This approach is extremely popular.

- For example, in memetic algorithms [16], various kinds of local search are embedded in an evolutionary algorithm for locally improving candidate solutions obtained from variation operators.
- *Very large scale neighborhood search* (VLSN) approaches are another example [17]. They utilize certain exact techniques such as dynamic programming to efficiently find best solutions in specifically designed large neighborhoods within a local search based metaheuristic.
- Also, any decoder-based metaheuristic, in which a master algorithm acts on an implicit or incomplete representation of candidate solutions and a decoder is used to obtain corresponding actual solutions, falls into this category. Such a decoder can be virtually any kind of algorithm ranging from a simple problem specific heuristic to sophisticated exact optimization techniques or other OR/AI methods. For example in the cutting and packing domain, a common approach is to represent a candidate solution as a permutation of the items that need to be cut out or packed, and an actual solution is derived by considering the items in more or less sophisticated assignment heuristics in the given order, see e.g. [18]. Weight-coding [19] and problem space search [20] are further examples of indirect, relatively generally applicable representations based on decoders.
- Merging solutions: In population based methods such as evolutionary algorithms or scatter search, a traditional variation operator is recombination.

It derives a new solution by combining features of two (or more) parent solutions. Especially in classical genetic algorithms, this operator is based on pure random decisions and therefore works without exploiting any problem specific knowledge. Occasionally, this procedure is replaced by more powerful algorithms like path-relinking [21] or by exact techniques based on branch-and-bound or integer linear programming that identify a best combination of parental features, see e.g. [22,23].

In collaborative combinations, algorithms exchange information, but are not part of each other. For example, the popular island model [24] for parallelizing evolutionary algorithms falls into this category. We can further classify the traditional island model as a homogeneous approach since several instances of the same metaheuristic are performed. In contrast, Talukdar et al. [25,26] suggested a heterogeneous framework called *asynchronous teams* (A-Teams). An A-Team is a problem solving architecture consisting of a collection of agents and memories connected into a strongly cyclic directed network. Each of these agents is an optimization algorithm and can work on the target problem, on a relaxation of it, i.e. a superclass, or on a subclass. The basic idea of A-Teams is having these agents work asynchronously and autonomously on a set of shared memories. Denzinger and Offermann [27] presented a similar multi-agent based approach for achieving cooperation between search-systems with different search paradigms, such as evolutionary algorithms and branch-and-bound.

In particular in collaborative combinations, a further question is which search spaces are actually explored by the individual algorithms. According to [14] we can distinguish between an implicit decomposition resulting from different initial solutions, different parameter values etc., and an explicit decomposition in which each algorithm works on an explicitly defined subspace. Effectively decomposing large problems is in practice often an issue of crucial importance. Occasionally, problems can be decomposed in very natural ways, but in most cases finding an ideal decomposition into relatively independent parts is difficult. Therefore, (self-)adaptive schemes are sometimes also used.

3 A Unified View on Hybrid Metaheuristics

The success of all these hybrid metaheuristics tells us that it is usually a bad idea to approach a given (combinatorial) optimization problem with a view that is too restricted to a small (sub-)class of metaheuristics, at least when the primary goal is to solve the problem as well as possible. There is nothing to say against the analogy to real-world phenomena, by which several metaheuristics are explained with or even derived from, for example evolutionary algorithms, ant colony optimization, or simulated annealing. However, one should avoid to focus too strongly on such philosophies, hereby losing the view on particular strengths and benefits of other algorithmic concepts.

Instead of perceiving the various well-known metaheuristics as relatively independent optimization frameworks and occasionally considering hybridization

Algorithm Pool Template
Initialize pool P by an external procedure;
while termination=FALSE **do**
  $S \leftarrow OF(P)$;
  if $|S| > 1$ **then**
    $S' \leftarrow SCM(S)$
  else
    $S' \leftarrow S$;
  $S'' \leftarrow IM(S')$;
  $P \leftarrow IF(S'')$;
Apply a post-optimizing procedure to P.

Fig. 2. The pool template from Voß [30,31]. P: Pool; IF/OF: Input/Output Function; IM: Improvement Method; SCM: Solution Combination Method.

for achieving certain benefits, it might be advantageous to change the point-of-view towards a unified design concept. All the existing metaheuristics share some ideas and differ among each other by certain characteristic *key components*. Making these key components explicit and collecting them yields a *toolbox* of components from which we can choose in the design of an optimization algorithm as it seems to be most appropriate for the target problem at hand.

In fact, this unified point-of-view is not new. Vaessens et al. [28] already presented a template for representing various kinds of local search based approaches, in particular threshold algorithms, tabu search, variable depth search, and even population based methods such as genetic algorithms. They also addressed multi-level approaches such as genetic local search, where a local search algorithm is applied within a genetic algorithm.

Calégary et al. [29] provided a taxonomy and united view on evolutionary algorithms and exemplarily discussed them with genetic algorithms, ant colony optimization, scatter search, and an emergent colonization algorithm.

Greistorfer and Voß [30,31] introduced a pool template by which they intend to cover even more different classes of metaheuristics, but especially also population based approaches. It is shown in Figure 2 and follows the definition of metaheuristics as given by Voß in [5] and cited in Section 1. To interpret, for example, simulated annealing in terms of this template, we set $|S| = 1$ and $|P| = 2$. The latter choice seems to be unusual at first glace. However, it covers the fact that we always have a current solution in the pool for which one or more neighbors are evaluated and additionally store the overall so-far best solution. The output function OF always simply returns the current solution. The improvement method IM includes the random choice of a neighboring solution and its evaluation, while the input function IF finally applies the Metropolis criterion (or some other condition) in order to either accept the new solution or to retain the previous one. The temperature update can also be considered to be part of the input function. Obviously, also other derivatives of local search like tabu search, guided local search, iterated local search, variable neighborhood descent/search, but also population-based approaches such as genetic algorithms,