A BRIEF SURVEY ON HYBRID METAHEURISTICS

Christian Blum
ALBCOM research group
Universitat Politcnica de Catalunya, Barcelona, Spain
cblum@lsi.upc.edu

Jakob Puchinger
Mobility Department
Austrian Institute of Technology, Vienna, Austria
jakob.puchinger@ait.ac.at

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

Andrea Roli
Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)
Alma Mater Studiorum, Università di Bologna, Campus of Cesena, Italy
andrea.roli@unibo.it

Abstract The combination of components from different algorithms is currently one of the most successful trends in optimization. The hybridization of metaheuristics such as ant colony optimization, evolutionary algorithms, and variable neighborhood search with techniques from operations research and artificial intelligence plays hereby an important role. The resulting hybrid algorithms are generally labelled hybrid metaheuristics. The rising of this new research field was due to the fact that the focus of research in optimization has shifted from an algorithm-oriented point of view to a problem-oriented point of view. In this brief survey on hybrid metaheuristics we provide an overview on some of the most interesting and representative developments.

Keywords: Optimization, hybridization, metaheuristics, exact techniques

1. Introduction

The term *metaheuristic* was introduced to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem. Genetic and evolutionary algorithms, tabu search, simulated annealing, iterated local search, and ant colony optimization, just to name a few, are typical representatives falling under this generic term. Each of them has an individual historical background, follows certain paradigms and philosophies, and puts one or more particular strategic concepts in the foreground. For a detailed introduction to metaheuristics we refer the interested reader to [25, 12].

In contrast to the early days of metaheuristic research, the last 5-10 years have produced a large number of algorithms that simply do not fit into a single metaheuristic category. This is because these untraditional approaches combine various algorithmic ideas, often originating from several branches of artificial intelligence, operations research and computer science in general. Such approaches are commonly referred to as *hybrid metaheuristics* [10]. The lack of a precise definition of this term is sometimes subject to criticism. In our opinion, however, the relatively open nature of this term is rather helpful, as strict borderlines between related fields of research are often a hindrance for creative thinking and the exploration of new research directions.

The main motivation for the hybridization of different algorithmic concepts has been to obtain better performing systems that exploit and combine advantages of the individual pure strategies, that is, hybrids are believed to benefit from *synergy*. In fact, choosing an adequate combination of multiple algorithmic concepts is often the key for achieving top performance in solving many hard optimization problems. However, the task of developing a highly effective hybrid approach is not easy at all. Nevertheless, there are several hybridization types that have proven successful on many occasions, and they can provide some guidance.

The growing popularity of this line of research is documented by rather recent conferences and workshops such as *CPAIOR* [62], *Hybrid Metaheuristics* [6], and *Matheuristics* [38]. Moreover, the first book specifically devoted to hybrid metaheuristics has recently been published in 2008 [10]. In this brief survey, we provide an overview of hybrid metaheuristics by illustrating prominent and paradigmatic examples, which range from the integration of metaheuristic techniques among themselves, to the hybridization of metaheuristics with constraint and mathematical programming. The interested reader can find other reviews on hybrid metaheuristics in [16, 19, 50, 10].

2. Examples and Literature Overview

In our opinion, the current body of research on hybrid metaheuristics can be subdivided into five different categories, namely, the hybridization of metaheuristics with (meta-)heuristics, constraint programming, tree search methods, problem relaxations, and dynamic programming. Each of these five categories is treated below in its own subsection. For each category we will list representative works.

2.1 Hybridization of Metaheuristics With (Meta-)Heuristics

The hybridization of metaheuristics with (meta-)heuristics is quite popular, especially for what concerns the use of local search methods inside population-based methods. Indeed, most of the successful applications of evolutionary computation and ant colony optimization make use of local search procedures for refining the generated solutions. This is because the major strength of population-based methods is their exploration capability. At the start of the search they generally try to capture a global picture of the search space, and typically, rather simple and problem-dependent operations are then iteratively applied to derive diverse new solutions successively focusing the search on promising regions of the search space. Conversely, the strength of local search methods is their rather fast intensification capability, that is, the capability of quickly finding better solutions in the vicinity of given starting solutions. In summary, population-based methods are good in identifying promising areas of the search space in which local search methods can then quickly determine the best solutions. Therefore, this type of hybridization is often very successful. In the field of evolutionary algorithms these hybrids even carry their own name, *memetic algorithms* [34].

Apart from the usual above-mentioned hybridization, this category contains, for example, so-called *multi-level* techniques [64, 65]. They are heuristic frameworks with the potential of making the search process of a metaheuristic more effective and efficient. The basic idea of a multilevel technique is the one of coarsening a given problem instance. Then, the problem is solved on the coarsened instance and the obtained solution is transformed in order to obtain a solution to the original instance. *Variable fixing* strategies [45, 49] are related techniques.

Another hybrid in this first category are *hyper-heuristics* [13]. They work on a higher level than classical metaheuristics, in the sense that they do not directly operate on the search space of the problem under consideration. Instead they operate on a search space consisting of lower-level heuristics—or even metaheuristics—for the tackled problem. Hyper-heuristics are broadly concerned with selecting the right metaheuristic at any situation.

Interestingly, in recent years a few examples have appeared for the use of components from population-based methods within metaheuristics based on local search. One of these examples concerns *population-based iterated local search* [58] where iterated local search is extended from working on a single solution to working on a population that is managed in the style of evolution strategies. In a different example, Lozano and García-Martínez [36] advocate the use of an evolutionary algorithm as a perturbation technique within iterated local search, while Resende et al. [53] devise several versions of a hybrid algorithm based on GRASP and path relinking methodologies.

An important branch of hybridization is the enhancement of metaheuristics with additional techniques for improving run-time, results, or both. Montemanni and Smith [40] propose an algorithm to solve the frequency assignment problem that is based on tabu search. Hereby, tabu search is enhanced by *heuristic manipulation*, a mechanism based on the idea that adding constraints to a problem results in a search space reduction, which, in turn, may facilitate the solution of the problem.

2.2 Hybridization of Metaheuristics With Constraint Programming

Constraint programming (CP) is a programming paradigm that is build upon constraints and constraint solving [37]. CP is generally said to be particularly effective in finding feasible solutions to highly constrained problems. On the other side, metaheuristics are generally very effective in finding good-quality solutions to mildly constrained optimization problems while requiring a limited amount of computational resources. In turnm, metaheuristics are generally not very effective in tackling highly constrained problems, while CP alone usually does not achieve a particularily high performance in solving loosely constrained optimization problems. Given these considerations, the combination of metaheuristics with CP seems applicable to problems with a fairly high number of constraints and, at the same time, a sufficiently large number of feasible solutions [39].

The integration of (meta)heuristics and CP dates back to the late 1990s; see, for example, the works by Pesant and Gendreau [42, 43] and subsequent works [18, 57]. A survey on possible ways of integrating metaheuristics and CP is provided by Focacci et al. in [23]. With a bit of oversimplification, four main approaches for the integration of metaheuristics and CP can be identified:

1 Metaheuristics are applied before CP, providing a valuable input, or vice versa.

2 Metaheuristics, mainly local search methods, use CP to efficiently explore the neighborhood of the current solution.

3 Construction-based metaheuristics use CP in order to prune the search space.

4 CP applies a metaheuristic in order to improve a solution (i.e., a leaf of the search tree) or a partial solution (i.e., an inner node). Metaheuristic concepts can also be used to obtain incomplete but efficient tree exploration strategies.

The first one of these approaches represents a rather loose hybridization and can be seen as an instance of *cooperative search* [21]. The second approach combines the advantages of a fast search space exploration by means of a metaheuristic with the efficient neighborhood exploration performed by a systematic method. A prominent example of such a kind of integration is large neighborhood search and related approaches [56, 14]. The third approach has found applications especially in ant colony optimization [39, 33, 60]. Hereby, CP is used at each solution construction step to filter the available options for the extension of the current partial solution. The fourth approach preserves the search space exploration based on systematic search (such as tree search), but sacrifices the exhaustive nature of the search [27, 28]. The hybridization is usually achieved by integrating concepts developed in the context of metaheuristics (e.g., probabilistic choices, aspiration criteria, heuristic construction) into tree search methods. For example, instead of a chronological backtracking, a backjumping based on search history or information retrieved from local search samples can be performed. Other examples of this approach can be found in [55, 47].

2.3 Hybridizing Metaheuristics with Tree Search

Optimization techniques can be characterized by their way of exploring the search space. Some algorithms consider the search space of an optimization problem in form a of a tree, the so-called *search tree*, which is generally defined by an underlying solution construction mechanism. Each path from the root node of the search tree to one of the leafs corresponds to a step-by-step construction of a candidate solution. Inner nodes of the tree are partial solutions to the given problem. The process of moving from an inner node to one of its child nodes is called a solution construction step, or an extension of a partial solution.

The class of tree search algorithms comprises approximate methods such as ant colony optimization and GRASP, but also complete techniques such as branch & bound and heuristic variants of complete techniques such as beam search. Therefore, the hybridization of metaheuristics with other tree search techniques is probably one of the most popular hybridization approaches. One of the first works on a combination of branch & bound with an evolutionary algorithm is the one by Nagar et al [41]. Hereby, an incomplete execution of branch & bound is used to guide the working of an evolutionary algorithm.

Exact tree search methods have been used quite a few times in *solution merging*, which is based on the idea of deriving new and hopefully better solutions from the attributes originating from two ore more input solutions. Applegate et al. [2, 3] were among the first to apply tree search methods in the context of merging. In an application for the travelling salesman problem they merge solutions and produce a (potentially) new solution by solving the resulting reduced graph to optimality.

Similarly as constraint programming is sometimes used for searching large neighborhoods (see Section 2.2), other tree search methods are also utilized for this purpose. Especially branch & bound techniques based on linear programming, including branch-and-cut, are often a promising option when the problem at hand can be expressed by a mixed integer programming (MIP) model. The availability of highly effective general purpose MIP solvers, which are typically based on sophisticated branch-and-cut frameworks but nevertheless can be relatively easily applied, makes this approach particularly interesting in practice. In the literature, numerous successful examples exist for such approaches. Among the more generally applicable ones is *local branching* [22]. A successful problem-specific example for large neighborhood search by means of solving sub-MIPs via branch-and-cut has been described by Prandtstetter and Raidl for the car sequencing problem [46].

Instead of using an exact method within a metaheuristics, the literature also offers examples where incomplete versions of exact methods are used to enhance metaheuristics. One of these examples is Beam-ACO [7, 8], which is a hybrid algorithm that combines ant colony optimization with beam search. This algorithm employs parallel and nonindependent solution constructions at each iteration, in the style of beam search. Another example is the hybridization of metaheuristics with backtracking. In [29] the authors describe applications of various hybrid metaheuristics to problems ranging from car sequencing and graph coloring to scheduling. For example, a tabu search algorithm for the job shop scheduling problem is presented, combining local search with complete enumeration as well as limited backtracking search.

The examples mentioned above are characterized by a subordinate use of an exact method within the metaheuristic. However, the literature also offers examples where metaheuristics are used for guiding the search process of an exact technique, or a heuristic derivate. For example, Rothberg [54] suggests a tight integration of an evolutionary algorithm in a MIP solver based on branch & cut. The evolutionary algorithm is applied at regular intervals as a branch & bound tree node heuristic. Another example concerns the works presented in [24, 11], where the applications of beam search and a memetic algorithm are intertwined. More specifically, phases of beam search and the memetic algorithm alternate. Beam search purges its queue of open partial solutions by excluding those ones whose upper bounds are worse than the value of the best solution found by the memetic algorithm. On the other side, beam search guides the search of the memetic algorithm by injecting information about promising regions of the search space into the population. Another example where metaheuristics may be used as a subordinate technique is *diving* [17], which is a mechanism for focusing the search process of branch & bound in an initial phase to neighborhoods of promising incumbents in order to quickly identify high-quality solutions.

2.4 Hybridization of Metaheuristics With Problem Relaxation

Guiding metaheuristics by *problem relaxation* has become quite popular in recent years. A so-called relaxed problem is obtained by simplifying or omitting constraints from the original problem formulation. The hope is, first, that the relaxed problem can be efficiently solved, and second, that the structure of an optimal solution to the relaxed problem together with its objective function value can be used in some way for solving the original problem. For example, the optimal solution value of a relaxed problem can be seen as a bound for the optimal solution value of the original problem. Therefore, it can be used in a branch & bound algorithm for discarding parts of the search tree. An important type of relaxation in combinatorial optimization concerns dropping the integrality constraints of the involved variables from a MIP formulation. The resulting linear programming (LP) relaxation can then be solved to optimality by efficient methods like the well-known *Simplex algorithm*.

One of the most obvious ways to utilize an optimal solution to the LP relaxation of a problem at hand is to directly derive a heuristic integer solution which is feasible for the original problem. Depending on the problem, this can be achieved by simple rounding or more sophisticated repairing strategies. For example, Raidl and Feltl [51] present a hybrid genetic algorithm for the generalized assignment problem. The LP relaxation of the problem is solved and its solution is exploited by a randomized rounding procedure to create an initial population of promising integral solutions.

In [63], Vasquez and Hao present a two-phase approach for the multidimensional 0-1 knapsack problem (MKP). This algorithm is a prime example for a hybrid metaheuristic guided by problem relaxation. The main idea consists in solving a number of relaxed problems obtained by dropping the integrality constraints to optimality. This is done in a first phase. Afterwards, in a second phase, tabu search is used to search around the optimal solutions to the relaxed problems. Another examples concerns the work by Puchinger and Raidl [48]. They introduced *relaxation guided variable neighborhood search* (RGVNS). The main algorithmic framework of RGVNS is variable neighborhood search. However, neighborhoods are dynamically ordered according to so-called *improvement-potentials*. These estimates are determined by computing bounds on the objective function values of the optimal solutions within each neighborhood. Such bounds are obtained by solving a relaxation of the original problem.

A successful example for using other relaxation techniques is the hybrid Lagrangian GA for the prize collecting Steiner tree problem by Haouari and Siala [26]. Hereby, the GA uses results from the previously solved Lagrangian relexation. In particular, the original graph is reduced by discarding edges, meaningful initial solutions are generated, and the objective function is modified by considering reduced costs.

For the knapsack constrained maximum spanning tree problem, a similar combination of Lagrangian decomposition and a genetic algorithm is described in Pirkwieser et al. [44]. A combination of a Lagrangian relaxation approach and a variable neighborhood descent metaheuristic, which is also based on similar principles, has recently been developed for a real-world fiber optic network design problem by Leitner and Raidl [35].

Tamura et al. [59] propose an algorithm that works by first executing a GA for identifying a promising region of the search space. The fitness of solutions is hereby related to Lagrangian relaxations. Afterwards, an exhaustive search is used to find the best solution within the identified region.

Finally, Reimann [52] introduces an ant colony optimization algorithm for the symmetric travelling salesman problem where an optimal solution to the minimum spanning tree relaxation is used for biasing the search of the artificial ants towards edges that form part of the minimum spanning tree.

3. Hybridization of Metaheuristics With Dynamic Programming

Dynamic programming (DP) is another example of an optimization method from operations research and control theory that can be successfully integrated with metaheuristics, both in the case of constructive and local search techniques. DP provides a method for defining an optimal strategy that leads from an initial state to the final goal and it has been successfully applied to many optimization and control problems [5].

Iterated dynasearch is a hybrid metaheuristic that uses DP as a neighborhood exploration strategy inside iterated local search [30]. The rationale behind this integration is that in neighborhood search, the larger the neighborhood size, the better the quality of the local optimum returned (on average). Suitable neighborhoods are often of exponential size, making it impractical to perform an explicit exhaustive lexicografical enumeration. Therefore, more computationally efficient neighborhood exploration techniques are required. In some cases, DP can make it possible to completely explore an exponential size neighborhood in polynomial time and space [15, 1].

In [9], Blum and Blesa present the use of a DP algorithm within two different metaheuristics for the k-cardinality tree (KCT) problem. The general idea of their approaches is not limited to the KCT problem and can, potentially, be used for other subset problems. Basically, the idea is to let the metaheuristic generate objects that are bigger than solutions, containing in general an exponential number of solutions to the problem under consideration. DP is then used to efficiently find for each object the best solution that it contains.

Hu and Raidl [31] use DP within an evolutionary algorithm as a mechanism for generating the best solution that can be obtained from an incomplete solution. A somewhat related approach is presented in [20]. In [32], DP is used purely as a decoder for tackling the rectangle packing problem with general spatial costs, which consists in packing given rectangles without overlap in the plane so that the maximum cost of the rectangles is minimized.

The following examples deal with hybridizations based on problem decompositions. In [66], the authors propose a hybrid method that combines adaptive memory, sparse DP, and reduction techniques to reduce and explore the search space. The first step consists in the generation of a bi-partition of the variables. The resulting small problem is solved using the forward-phase of DP. The space defined by the remaining variables is explored using tabu search. Hereby, each partial solution is completed with the information stored during the forward phase of DP. The application of DP to subproblems is also proposed in [61]. The presented local search technique called *iterative dynamic programming* works by subdividing the problem into subproblems, and optimizing the subproblems seperately by DP.

Finally, we would like to point out an interesting heuristic version of DP known as *bounded dynamic programming* in which at each level the number of states is heuristically reduced. In this way, the authors of [4] were able to find most optimal solutions to benchmark instances of the simple assembly line balancing problem in a reduced amount of computation time.

4. Conclusions

Research on hybrid metaheuristics is still in its early stages. However, we are convinced that, in the years to come, most publications on metaheuristic applications will be concerned with hybrids. Nevertheless, the process of designing and implementing hybrid metaheuristics is rather complicated and involves knowledge about a broad spectrum of algorithmic techniques, programming and data structures, as well as algorithm engineering and statistics. In fact, it is hardly possible to provide guidelines for the successful development of hybrid metaheuristics. However, in the process of developing a hybrid metaheuristic it is indispensible (1) to carefully search the literature for the most successful optimization approaches for the problem at hand or for similar problems, and (2) the study of different ways of combining the most promising features of the identified approaches. We hope that this paper may serve as a starting point for this purpose.

References

[1] E. Angel and E. Bampis. A multi-start dynasearch algorithm for the time dependent single-machine total weighted tardiness scheduling problem. *European Journal of Operational Research*, 162(1):281–289, 2005.

[2] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. On the solution of the traveling salesman problem. *Documenta Mathematica*, Extra Volume ICM III:645–656, 1998.

[3] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.

[4] J. Bautista and J. Pereira. A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research*, 194(3):787–794, 2009.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua, NH, 3rd edition, 2007.

[6] M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, and A. Schaerf, editors. *Proceedings of HM 2009 – Sixth International Workshop on Hybrid Metaheuristics*, volume 5818 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2009.

[7] C. Blum. Beam-ACO–hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Computers and Operations Research*, 32:1565–1591, 2005.

[8] C. Blum. Beam-ACO for simple assembly line balancing. *INFORMS Journal on Computing*, 20(4):618–627, 2008.

[9] C. Blum and M. J. Blesa. Solving the KCT problem: Large-scale neighborhood search and solution merging. In E. Alba, C. Blum, P. Isasi, C. León, and J. A. Gómez, editors, *Optimization Techniques for Solving Complex Problems*, pages 407–421. Wiley & Sons, Hoboken, NJ, 2009.

[10] C. Blum, M. J. Blesa Aguilera, A. Roli, and M. Sampels, editors. *Hybrid Metaheuristics – An Emerging Approach to Optimization*, volume 114 of *Studies in Computational Intelligence*. Springer, 2008.

[11] C. Blum, C. Cotta, A. J. Fernández, J. E. Gallardo, and M. Mastrolilli. Hybridization of metaheuristics with branch & bound derivatives. In Blum et al. [10], pages 85–116.

[12] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[13] E. K. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyperheuristics: An emerging direction in modern search technology. In Glover and Kochenberger [25], pages 457–474.

[14] Y. Caseau and F. Laburthe. Effective Forget-and-Extend heuristics for scheduling problems. In *Proceedings of CP-AI-OR'02 – Fourth Int. Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems*, 1999.

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

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

[17] E. Danna, E. Rothberg, and C. Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming, Series A*, 102:71–90, 2005.

[18] B. De Backer, V. Furnon, and P. Shaw. Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics*, 6:501–523, 2000.

[19] I. Dumitrescu and T. Stuetzle. Combinations of local search and exact algorithms. In Günther R. Raidl et al., editors, *Applications of Evolutionary Computation*, volume 2611 of *Lecture Notes in Computer Science*, pages 211–223. Springer, Berlin, Germany, 2003.

[20] T. Dunker, G. Radons, and E. Westkämper. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research*, 165(1):55–69, 2005.

[21] Mohammed El-Abd and Mohamed Kamel. A taxonomy of cooperative search algorithms. In M. J. Blesa Aguilera, C. Blum, A. Roli, and M. Sampels, editors, *Proceedings of HM 2005 – Second International Workshop on Hybrid Metaheuristics*, volume 3636 of *Lecture Notes in Computer Science*, pages 32–41. Springer Verlag, Berlin, Germany, 2005.

[22] M. Fischetti and A. Lodi. Local Branching. *Mathematical Programming, Series B*, 98:23–47, 2003.

[23] F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In Glover and Kochenberger [25], pages 369–403.

[24] J. E. Gallardo, C. Cotta, and A. J. Fernández. On the hybridization of memetic algorithms with branch-and-bound techniques. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 37(1):77–83, 2007.

[25] F. Glover and G. Kochenberger, editors. *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 2003.

[26] M. Haouari and J. C. Siala. A hybrid Lagrangian genetic algorithm for the prize collecting Steiner tree problem. *Computers & Operations Research*, 33(5):1274–1288, 2006.

[27] W. D. Harvey. *Nonsystematic Backtracking Search*. PhD thesis, CIRL, University of Oregon, Eugene, Orgeon, 1995.

[28] W. D. Harvey and M. L. Ginsberg. Limited discrepancy search. In C. S. Mellish, editor, *Proceedings of IJCAI 1995 – 14th International Joint Conference on Artificial Intelligence*, volume 1, pages 607–615. Morgan Kaufmann Publishers, San Mateo, CA, 1995.

[29] P. Van Hentenryck and L. Michel. *Constraint-Based Local Search*. MIT Press, Cambridge, MA, 2005.

[30] H. Hoos and T. Stützle. *Stochastic Local Search – Foundations and Applications*. Morgan Kaufmann Publishers, 2005.

[31] B. Hu and G. R. Raidl. Effective neighborhood structures for the generalized traveling salesman problem. In J. I. van Hemert and C. Cotta, editors, *Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2008*, volume 4972 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2008.

[32] S. Imahori, M. Yagiura, and T. Ibaraki. Improved local search algorithms for the rectangle packing problem with general spatial costs. *European Journal of Operational Research*, 167(1):48–67, 2005.

[33] M. Khichane, P. Albert, and C. Solnon. Integration of ACO in a constraint programming language. In *Ant Colony Optimization and Swarm Intelligence, 6th International Workshop, ANTS 2008*, volume 5217 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2008.

[34] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.

[35] M. Leitner and G. R. Raidl. Lagrangian decomposition, metaheuristics, and hybrid approaches for the design of the last mile in fiber optic networks. pages 158–174.

[36] M. Lozano and C. García-Martínez. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research*, 2009. In press.

[37] K. Marriott and P. J. Stuckey. *Introduction to Constraint Logic Programming*. MIT Press, Cambridge, MA, 1998.

[38] Matheuristics 2008. http://astarte.csr.unibo.it/matheuristics2008/. Visited in April 2009.

[39] B. Meyer. *Hybrids of constructive meta-heuristics and constraint programming: A case study with ACO*, chapter 6. Volume 114 of Blum et al. [10], 2008.

[40] R. Montemanni and D. H. Smith. Heuristic manipulation, tabu search and frequency assignment. *Computers & Operations Research*, 2009. In press.

[41] A. Nagar, S. S. Heragu, and J. Haddock. A meta-heuristic algorithm for a bi-criteria scheduling problem. *Annals of Operations Research*, 63:397–414, 1995.

[42] G. Pesant and M. Gendreau. A view of local search in Constraint Programming. In *Principles and Practice of Constraint Programming - CP'96*, volume 1118 of *Lecture Notes in Computer Science*, pages 353–366. Springer-Verlag, 1996.

[43] G. Pesant and M. Gendreau. A Constraint Programming Framework for Local Search Methods. *Journal of Heuristics*, 5:255–279, 1999.

[44] S. Pirkwieser, G. R. Raidl, and J. Puchinger. Combining Lagrangian decomposition with an evolutionary algorithm for the knapsack constrained maximum spanning tree problem. In C. Cotta and J. I. van Hemert, editors, *Evolutionary Computation in Combinatorial Optimization – EvoCOP 2007*, volume 4446 of *Lecture Notes in Computer Science*, pages 176–187. Springer, Berlin, Germany, 2007.

[45] D. Pisinger. Core problems in knapsack algorithms. *Operations Research*, 47:570–575, 1999.

[46] M. Prandtstetter and G. R. Raidl. An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem. *European Journal of Operational Research*, 191(3), 2008.

[47] S. Prestwich. Combining the scalability of local search with the pruning techniques of systematic search. *Annals of Operations Research*, 115:51–72, 2002.

[48] J. Puchinger and G. R. Raidl. Bringing order into the neighborhoods: Relaxation guided variable neighborhood search. *Journal of Heuristics*, 14(5):457–472, 2008.

[49] J. Puchinger, G. R. Raidl, and U. Pferschy. The core concept for the multidimensional knapsack problem. In Jens Gottlieb and Günther R. Raidl, editors, *Evolutionary Computation in Combinatorial Optimization – EvoCOP 2006*, volume 3906 of *Lecture Notes in Computer Science*, pages 195–208. Springer, Berlin, Germany, 2006.

[50] G. R. Raidl. A unified view on hybrid metaheuristics. In F. Almeida, M. J. Blesa Aguilera, C. Blum, J. M. Moreno Vega, M. Pérez Pérez, A. Roli, and M. Sampels, editors, *Proceedings of HM 2006 – Third International Workshop on Hybrid Metaheuristics*, volume 4030 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, Berlin, Germany, 2006.

[51] G. R. Raidl and H. Feltl. An improved hybrid genetic algorithm for the generalized assignment problem. In H. M. Haddadd et al., editors, *Proceedings of*

the 2003 ACM Symposium on Applied Computing, pages 990–995. ACM Press, 2004.

[52] M. Reimann. Guiding ACO by problem relaxation: A case study on the symmetric TSP. In T. Bartz-Beielstein, M. J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Proceedings of HM 2007 – Fourth International Workshop on Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 45–55. Springer Verlag, Berlin, Germany, 2007.

[53] M. G. C. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path relinking for the maxmin diversity problem. *Computers & Operations Research*, 2009. In press.

[54] E. Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS Journal on Computing*, 19(4):534–541, 2007.

[55] A. Schaerf. Combining local search and look-ahead for scheduling and constraint satisfaction problems. In *Proceedings of IJCAI 1997 – 15th International Joint Conference on Artificial Intelligence*, pages 1254–1259. Morgan Kaufmann Publishers, San Mateo, CA, 1997.

[56] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.-F. Puget, editors, *Principle and Practice of Constraint Programming – CP98*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer Verlag, Berlin, Germany, 1998.

[57] P. Shaw, B. De Backer, and V. Furnon. Improved local search for CP toolkits. *Annals of Operations Research*, 115:31–50, 2002.

[58] T. Stützle. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174(3):1519–1539, 2006.

[59] H. Tamura, A. Hirahara, I. Hatono, and M. Umano. An approximate solution method for combinatorial optimisation. *Transactions of the Society of Instrument and Control Engineers*, 130:329–336, 1994.

[60] D. R. Thiruvady, C. Blum, B. Meyer, and A. T. Ernst. Hybridizing Beam-ACO with constraint programming for single machine job scheduling. In M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, and A. Schaerf, editors, *Proceedings of HM 2009 – Sixth International Workshop on Hybrid Metaheuristics*, volume 5818 of *Lecture Notes in Computer Science*, pages 30–44. Springer Verlag, Berlin, Germany, 2009.

[61] S.-M. Tse, Y. Liang, K.-S. Leung, K.-H. Lee, and T. S.-K. Mok. A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 37(1):84–91, 2007.

[62] W. J. van Hoeve and J. N. Hooker, editors. *Proceedings of CPAIOR 2009 – 6th International Conference on the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5547 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2009.

[63] M. Vasquez and Y. Vimont. Improved results on the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 165(1):70–81, 2005.

[64] C. Walshaw. Multilevel refinement for combinatorial optimization problems. *Annals of Operations Research*, 131:325–372, 2004.

[65] C. Walshaw. Multilevel refinement for combinatorial optimisation: Boosting metaheuristic performance. In Blum et al. [10], pages 261–289.

[66] C. Wilbaut, S. Hanafi, A. Fréville, and S. Balev. Tabu search: global intensification using dynamic programming. *Control and Cybernetics*, 35(3):579–598, 2009.