# Matheuristics: Survey and Synthesis

Marco A. Boschetti* Adam N. Letchford† Vittorio Maniezzo‡

To appear in *International Transactions in Operational Research*

**Abstract**

In integer programming and combinatorial optimisation, people use the term *matheuristics* to refer to methods that are heuristic in nature, but draw on concepts from the literature on exact methods. We survey the literature on this topic, with a particular emphasis on matheuristics that yield both primal and dual bounds (i.e., upper and lower bounds in the case of a minimisation problem). We also make some comments about possible future developments.

**Keywords:** integer programming, combinatorial optimisation, heuristics, matheuristics, Lagrangian relaxation, dual ascent.

## 1 Introduction

Ever since the 1960s, integer programming and combinatorial optimisation problems have received much attention from mathematicians, computer scientists and operational researchers, due to the huge range of important practical applications (see, e.g., [62, 128, 189]). Unfortunately, many problems of interest are NP-hard (see [9]), which means that large-scale instances can be very challenging to solve.

In this context, a key distinction is between *exact* and *heuristic* methods. Exact methods are guaranteed to solve instances to proven optimality, given enough computing resources (i.e., time and memory). Heuristics are not guaranteed to find optimal solutions, but they tend to be faster, and they often yield solutions which are of "acceptable" quality. Although significant progress has been made in exact methods (e.g., [62, 189]), heuristics remain extremely useful in many cases (e.g., [147, 163]).

*Department of Mathematics, University of Bologna, via dell'Università 50, Cesena, 47521, Italy. E-mail: `marco.boschetti@unibo.it`

†Department of Management Science, Lancaster University, Lancaster LA1 4YX, UK. E-mail: `a.n.letchford@lancaster.ac.uk`

‡Department of Computer Science and Engineering, University of Bologna, via dell'Università 50, Cesena, 47521, Italy. E-mail: `vittorio.maniezzo@unibo.it`

Many authors use the term *metaheuristics* to refer to general-purpose heuristic frameworks, such as Simulated Annealing and Tabu Search (see the textbooks [147, 163]). More recently, researchers started using the term *matheuristics* to refer to (meta-)heuristics that draw on concepts from the traditional mathematical programming literature (see [144, 178]). Such heuristics may have subroutines that involve, for example, linear programming, integer programming, dynamic programming, Lagrangian relaxation, or Benders decomposition.

The first international matheuristics conference took place in Bertinoro, Italy in 2006 (see [42]). Ever since then, the field of matheuristics has been in rapid expansion. For details, we refer the reader to the surveys [18, 42, 79, 165, 166] and the books [143, 144].

In this new survey, we approach the topic from a different viewpoint, placing a particular emphasis on matheuristics that yield both *primal* and *dual* bounds. For a minimisation problem, such matheuristics yield not only a feasible integer solution (with an associated upper bound), but also the solution to some kind of relaxed problem or dual problem (with an associated lower bound). In our view, matheuristics of this kind are highly desirable, since the bounds help one to evaluate the performance of the method with higher precision, on any given instance.

The paper has the following structure. Section 2 covers matheuristics that are based on *linear programming* (LP) relaxation and/or duality. Section 3 deals with ones based on *Lagrangian relaxation* (LR), *surrogate relaxation* (SR), or closely-related methods. Section 4 concerns heuristics based on *decomposition* techniques, such as Dantzig-Wolfe or Benders decomposition. Then, in Section 5, we mention a few popular matheuristics that do not necessarily yield dual bounds. Finally, in Section 6, we make some remarks about the strengths and weaknesses of each approach, and suggest some topics for future research.

Throughout, we write "ILP" for *integer linear program* and "COP" for *combinatorial optimisation problem.* We assume that the reader is familiar with the idea of formulating COPs as ILPs, along with the basics of integer programming (see, e.g., [62, 189]). For ease of notation, we assume that an ILP with $n$ variables and $m$ constraints takes the form

$$\min\left\{c^T x : \ Ax \geq b, \ x \in \mathbb{Z}^n_+\right\}, \tag{1}$$

where $c \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$. We use the convention that all vectors are column vectors. We sometimes also speak of *mixed-integer programs* (MIPs), by which we mean problems that are similar to ILPs, except that not all variables are required to take integer values.

## 2 Methods Based on LP Relaxation and Duality

In this section, we survey matheuristics that are based on LP relaxation and/or duality. Subsection 2.1 recalls the key concepts needed, such as relaxation, primal and dual pairs, and reduced costs. Subsections 2.2 and 2.3 concern matheuristics that use only primal or reduced-cost information, respectively. Subsection 2.4 covers matheuristics that explicitly use dual information.

### 2.1 Recap on LP relaxation and duality

If we relax the integrality constraint in the ILP (1), we obtain the following problem:

$$\min\left\{c^T x : \ Ax \geq b, \ x \in \mathbb{R}^n_+\right\}, \tag{2}$$

This is an LP, which is typically much easier to solve. Trivially, the solution to the LP gives a *lower bound* for the original COP.

The following facts can be found in any textbook on LP (e.g., [184]). The *dual* of the above LP is:

$$\max\left\{b^T y : \ A^T y \leq c, \ y \in \mathbb{R}^m_+\right\}. \tag{3}$$

The original LP is called the *primal*. The strong duality theorem states that, if $x^*$ and $y^*$ are optimal primal and dual solutions, respectively, then $c^T x^* = b^T y^*$. Moreover, the components of $y^*$ are the *dual prices* for the primal constraints. The vectors $s^* = Ax^* - b \in \mathbb{R}^m_+$ and $\rho^* = c - A^T y^* \in \mathbb{R}^n_+$ are called the *surplus* vector and the *reduced cost* vector, respectively. The conditions $(\rho^*)^T x^* = 0$ and $(s^*)^T y^* = 0$, called *complementary slackness*, always hold when $x^*$ and $y^*$ are optimal.

### 2.2 Heuristics that use primal information only

Among the many LP-based matheuristics, the easiest ones to understand are those that use primal information only. The intuition behind these methods is that an optimal (or near-optimal) LP solution $x^*$ is likely to contain some information that could be exploited by a heuristic. For example, if all variables in the ILP are binary, one might hope that variables with $x^*$-value close to 1 will have a high probability of taking the value 1 in optimal (or near-optimal) solutions to the ILP.

One strand of literature is concerned with the use of LP-based heuristics for specific COPs. Two good early examples are the "LP-rounding" heuristic for the set covering problem, due to Hochbaum [115], and the "randomised rounding" heuristic for network design problems, due to Raghavan and Tompson [164]. These two heuristics are also interesting because they are *approximation algorithms*, which means that they are guaranteed to produce solutions whose cost is within a known factor of the optimum. Other notable examples include an "iterative rounding" heuristic for shift scheduling, given in Thompson [180], and a similar heuristic for lot-sizing with setups, in Maes *et al.* [139]. More recently, LP solutions have been used as "seeds" for standard metaheuristic search [51].

A parallel strand of literature is concerned with LP-based matheuristics for ILPs in general. A good early example, from 1969, is the "interior-path" method of Hillier [114]. It starts at the LP optimum $x^*$, and then follows a path from $x^*$ to the interior of the feasible region. As it goes along, an attempt is made to find "nearby" integer solutions. For extensions to this method, see [78].

In 1980, Balas & Martin [17] devised an LP-based heuristic for 0-1 LPs, that they called "pivot-and-complement". The method starts at $x^*$, and then performs a sequence of primal simplex pivots that attempt to "push" fractional variables out of the basis. Later on, several researchers suggested improving pivot-and-complement by adding a tabu search phase at the end (e.g., [1, 136]).

In the coming paragraphs, we mention some of the matheuristics based on primal LP solutions that came out more recently.

**OCTANE.** "OCTANE" is a matheuristic for pure 0-1 LPs, developed in 2001 by Balas *et al.* [14]. The basic idea is as follows. First, we define a polyhedron, called the $n$-dimensional octahedron, that circumscribes the unit hypercube and has one facet for every vertex of the hypercube. We start at a basic optimal LP solution $x^*$, and then move from it in some chosen direction. Eventually, we "hit" a facet of the $n$-dimensional octahedron. We then check the corresponding vertex of the hypercube. If it is feasible for the original 0-1 LP, we have our desired heuristic solution. This procedure is repeated for several "promising" directions.

Although OCTANE performed reasonably well, it does not seem to have received much further attention. This may be because some of the alternative matheuristics mentioned below have tended to perform even better.

**Relax-and-fix.** "Relax-and-Fix" is a primal heuristic for general MIPs, that works by solving a sequence of simpler MIPs. To our knowledge, it was first defined explicitly in 1998 by Wolsey [189]. In its original format [33, 189], it worked as follows. First, the set of integer variables is partitioned into subsets, say $S_1, \ldots, S_k$. A simpler MIP is then solved, in which only the variables in $S_1$ are declared integer. The variables in $S_1$ are then permanently fixed at the values that they take in the solution to the simpler MIP. The variables in $S_2$ are then declared integer, and the resulting MIP is solved, and so on.

This approach, while at times effective, limits the search to one pass through the list of predefined subsets. A simple generalization is to make the subset choice dynamically adaptive, but other extensions are possible.

**Diving heuristics.** Diving heuristics [39] are a family of MIP heuristics that iteratively fix variables to integer values until a feasible MIP solution is obtained. They can be thought of as a heuristic for rapidly moving from a given node of a branch-and-bound tree to a "leaf" node. The name comes from the fact that they "dive" to a leaf node without any possibility of backtracking.

A well-known and highly effective diving heuristic is *relaxation induced neighborhood search* (RINS) [65]. When applied at a given branch-and-bound node, RINS compares the fractional LP solution at the node with the incumbent MIP solution. Typically, the two solutions will differ in the values of some variables. RINS tries to force the two solutions to agree on all variables, by fixing all variables that have the same values and letting the solver try to solve optimally the residual MIP problem, called the *sub-MIP*. The sub-MIP can in fact be quite large if too few variables were fixed, so its solution could potentially take a time comparable with that of the original problem. To remedy this, a limit on the computational resources available for optimization is usually imposed.

Diving heuristics are also often used within *branch-and-price* algorithms [172]. For more on branch-and-price, see Section 4.

**The feasibility pump.** The Feasibility Pump [80] was initially conceived as a tool for finding initia feasible solutions to very challenging MIPs, and in this capacity it is often included in general-purpose MIP solvers. In the context of matheuristics, however, the feasibility pump can also be used to bring to integer feasibility a fractional and/or LP-infeasible solution. Such solutions arise not only in branch-and-bound, but also in some other approaches, such as Lagrangian heuristics or destroy-and-fix approaches.

The feasibility pump builds on the observation that an integer feasible solution is coincident with its rounding. Formally, if $P$ is the LP polytope of the problem under study, an integer feasible solution $x^*$ corresponds to a point in $P$ such that $x^* = \tilde{x}$, where $\tilde{x}$ denotes the rounding of $x^*$, i.e., the possibly infeasible solution where each variable that must be integer is brought to the corresponding nearest integer.

The search for feasibility is based on the minimization of a function measuring the distance between the two solutions, $\Delta(x^*, \tilde{x})$. The search starts from a point $x^* \in P$ and its rounding $\tilde{x}$. If $\tilde{x}$ is feasible, we have found a feasible integer solution and we stop the search, Otherwise, we start a *pumping cycle*. This means that we solve the linear problem $\min \{\Delta(x, \tilde{x}) : x \in P\}$. This yields a new point $x^*$, which we round to obtain a new point $\tilde{x}$. If $\tilde{x}$ is feasible, we stop. Otherwise we perform another pumping cycle, and so on.

Figure 1 shows the evolution of $\Delta$ in case of a simple ILP, which started from an optimal LP value of 231.45 to eventually converge to a final feasible solution of cost 333.

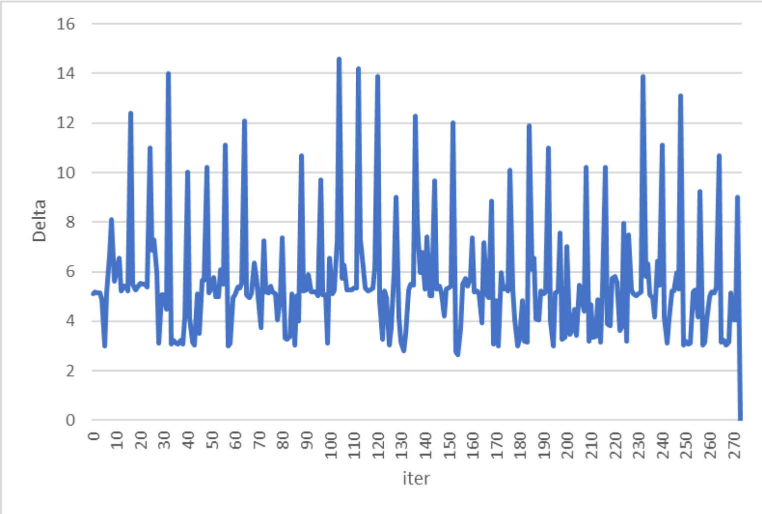


Figure 1: Feasibility pump: delta evolution

### 2.3 Heuristics that use reduced-cost information only

Now, recall from Subsection 2.1 the definition of the reduced-cost vector $\rho^*$. There are also matheuristics that rely only on $\rho^*$. The idea is that, if $\rho^*_j$ is very large, the variable $x_j$ is unlikely to appear in an optimal solution to the ILP. Conversely, if $\rho^*_j$ is zero or near-zero, there is a good chance that there is a (near-)optimal solution such that $x_j$ takes a positive integer value.

The first paper we found which mentions this idea explicitly is Mansini & Speranza [145], which is concerned with a portfolio selection problem. The problem is formulated as an MILP, and the LP relaxation is solved. In an attempt to obtain a good integer solution, a smaller MILP is solved, which contains all variables that have a positive value in the LP solution, plus the variables whose reduced cost is below some threshold. If time permits, a series of modified MILPs is solved, in which variables are added and dropped according to a heuristic rule.

The above heuristic inspired a general-purpose matheuristic for 0-1 LPs, called *Kernel Search* [6, 7]. Here is a brief overview of the approach. In the first phase, called the *initialisation* phase, the LP relaxation is solved, and a decision variable is called "promising" if it has a zero (or near-zero) reduced cost. The set of all promising variables forms the *kernel*. The remaining variables are partitioned into sets, called *buckets*, according to the value of their reduced cost. One then solves a simplified version of the 0-1 LP, in which only the variables in the kernel are present. In most cases, this yields a good feasible solution, with an accompanying upper bound.

If the gap between the lower and upper bounds is acceptable, the process terminates. Otherwise, the variables in the first bucket are added to the 0-1 LP, along with a constraint stating that at least one of the variables in the bucket must take the value 1. The simplified ILP is then re-solved. If the resulting solution is better than the previous one, the variables that are in the bucket and take the value of 1 in the solution are added to the kernel. If the gap between the lower and upper bounds is now acceptable, the process terminates. Otherwise, the variables in the second bucket are added to the 0-1 LP, and so on.

Kernel search has been applied with great success to the multi-dimensional knapsack problem [6], a portfolio selection problem [7], the capacitated facility location problem [107], and an inventory routing problem [8]. It has also been extended to general MILPs [106].

A variant of Kernel Search is the *Incremental Core* approach proposed in [44, 45]. The idea is to select a parameter $\rho^{\max}$, and include in the 0-1 LP only the variables with $\rho^*_j \leq \rho^{\max}$. If $\rho^{\max}$ is equal to the gap between the upper and lower bounds, the 0-1 LP solution is optimal. Otherwise, it is a heuristic solution. In the latter case, one can increase $\rho^{\max}$ in the hope of obtaining an improved heuristic solution (or even an optimal one).

Figure 2 shows a trace of an Incremental Core search on a resource-constrained project scheduling problem (from [151]) based on an "additive" lower bound (see Subsection 3.4). In this case, the problem could be solved to proven optimality. The plot shows the incremental contributions of the four successively computed bounds, together with the contribution of each new bound to the computation of improved feasible solutions.

## 2.4 Dual-based heuristics

When dealing with large-scale COP instances, even solving the initial LP relaxation (2) can be time-consuming. Fortunately, with the help of LP duality, one can compute valid lower bounds without explicitly solving the initial LP. Indeed, if $\bar{y}$ is *any* feasible solution to the dual LP (3), then $b^T\bar{y}$ is a lower bound for the original LP, and therefore for the original COP. Thus, if we wish to obtain a lower bound quickly, we can solve the dual approximately using some kind of heuristic. Moreover, as we will see, the dual solution can then be used to drive a matheuristic.

This idea first appeared in [37, 77], in the context of the *Uncapacitated Facility Location Problem* or UFLP. The authors of those papers proposed to start with all dual variables equal to zero, and then iteratively increase individual dual values until no further increases are possible. Perhaps surprisingly, this approach gives quite good lower bounds for many UFLP in-

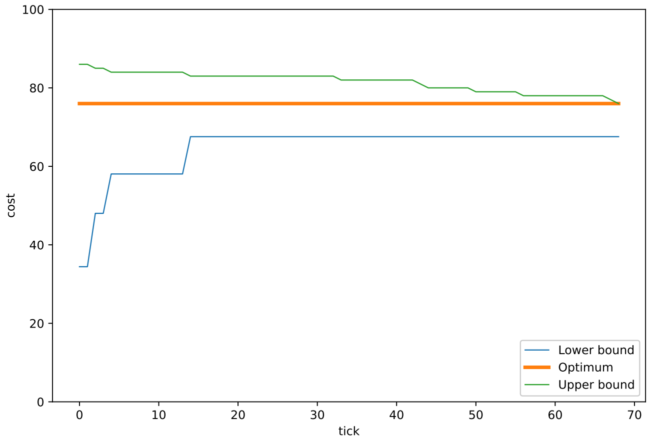


Figure 2: Incremental Core, upper and lower bound evolution

stances.

Erlenkotter [77] called this procedure *dual ascent*. He also proposed a simple local search procedure, called *dual adjustment*, which attempts to modify the current dual solution $y$ in order to improve the lower bound.

Now, for a given primal variable $x_j$ and a given dual solution $\bar{y}$, consider the following quantity:

$$\bar{c}_j(\bar{y}) \; = \; c_j - \sum_{i=1}^{m} \bar{y}_i A_{ij}.$$

If $\bar{y}$ is an optimal dual solution, then $\bar{c}_j(\bar{y})$ is the standard reduced cost, denoted by $\rho_j^*$ above. Even if $\bar{y}$ is not an optimal dual solution, however, it might still contain some useful information. In particular, one might expect variables with small $\bar{c}$ value to have a high probability of belonging to an optimal solution of the original ILP. Thus, the $\bar{c}$ values can be used within a heuristic.

The above observations led Erlenkotter to propose a primal heuristic for the UFLP, in which one iteratively opens facilities with zero $\bar{c}$ values, until certain "complementary slackness" conditions are met. The results were very encouraging, and the method was subsequently improved in [118, 127, 131].

Later on, dual ascent and dual adjustment were successfully applied to many other COPs, including (in roughly chronological order) the Steiner tree problem [161, 190], the multi-dimensional knapsack problem [6, 140], the set covering problem [13, 16, 26], the uncapacitated network design problem [12], the set partitioning problem [45, 91], the $p$-median problem [54], the design of railway networks [123], the multi-period assignment problem [156], the capacitated multi-commodity flow problem [21], the hub location problem [126], the quadratic assignment problem [112], and the fixed-charge transportation problem [49].

Some authors have considered more sophisticated matheuristics, which attempt to exploit primal and dual information in a more intelligent way. For brevity, we mention just a few examples. Fisher & Kedia [91] improved the approach to the set covering problem by using a local search phase to improve the dual solution, along with an improved primal heuristic. Wedelin [187] presented a primal-dual heuristic for a certain generalisation of the set partitioning problem arising in crew scheduling. The dual is solved heuristically via coordinate ascent. Later on, Hansen *et al.* [113] and Posta *et al.* [162] improved the dual ascent approach to the UFLP by incorporating sophisticated local search procedures for improving both primal and dual solutions.

To close this section, we mention that there is a significant literature on *primal-dual approximation algorithms* for NP-hard COPs. These algorithms iteratively build heuristic solutions to the primal and dual in parallel, in such a way that the gap between the corresponding upper and lower bounds is bounded in a specified way. For details, see the textbooks [185, 188].

# 3 Lagrangian and Surrogate Relaxation

In this section, we consider matheuristics that are based on *Lagrangian relaxation* (LR), *surrogate relaxation* (SR), or closely related methods. Subsection 3.1 recalls the basics of LR and SR. Subsections 3.2 and 3.3 review some LR-based and SR-based matheuristics. Subsection 3.4 mentions some related methods, such as Lagrangian dual ascent, semi-Lagrangian relaxation and additive bounding.

## 3.1 Recap on Lagrangian and surrogate relaxation

Some COPs of interest can be formulated as ILPs of the form

$$\min\left\{c^T x : \ Ax \geq b, \ Cx \geq d, \ x \in \mathbb{Z}^n_+\right\}, \tag{4}$$

where the constraints $Ax \geq b$ are "easy" and the constraints $Cx \geq d$ are "hard". By this, we mean that, if the "hard" constraints are dropped, the problem becomes significantly easier to solve.

Let us suppose that the number of "hard" constraints is $t$. In LR, we pick a vector $\lambda \in \mathbb{R}^t_+$ of *Lagrangian multipliers*, and then solve the following simpler ILP:

$$\min\left\{c^T x + \lambda^T(d - Cx) : \ Ax \geq b, \ x \in \mathbb{Z}^n_+\right\}. \tag{5}$$

Geoffrion [98] showed that, for any choice of $\lambda$, LR yields a lower bound for the original ILP. We will call this bound $L(\lambda)$.

The problem of finding the vector $\lambda$ which maximises $L(\lambda)$ is called the *Lagrangian dual* (LD). The LD is a piecewise-linear concave maximisation problem, and there exist several algorithms for solving it, either exactly or approximately (see, e.g., [109, 130]).

*Surrogate relaxation*, proposed in [99, 105], is similar to LR. The difference is that, instead of modifying the objective function in the ILP (4), we replace the complicating constraints $Cx \geq d$ with a single linear constraint. More precisely, for a given multiplier vector $\lambda$, we solve the following ILP:

$$\min \left\{ c^T x : \; Ax \geq b, \; (\lambda^T C)x \geq \lambda^T d, \; x \in \mathbb{Z}_+^n \right\}. \tag{6}$$

In most applications of SR, the system $Ax \geq b$ takes a very simple form. As a result, the ILP (6) is usually some kind of knapsack problem, and solved in pseudo-polynomial time by dynamic programming.

Let us denote by $S(\lambda)$ the lower bound obtained with SR. It is proved in [105] that $S(\lambda) \geq L(\lambda)$ for any given $\lambda$. The problem of finding the vector $\lambda$ which maximises $S(\lambda)$ is called the *surrogate dual* (SD). The SD is a quasi-concave maximisation problem, and is typically somewhat harder to solve than the LD (see, e.g., [72, 121, 125]).

### 3.2 Lagrangian heuristics

For a given multiplier vector $\lambda$, let $\bar{x}(\lambda)$ be the solution to the relaxed problem (5). By definition, $\bar{x}(\lambda)$ is integer and satisfies the constraints $Ax \geq b$, but it might fail to satisfy the constraints $Cx \geq d$. For some specific families of ILPs, it is possible to "repair" $\bar{x}(\lambda)$, with not much effort, in order to obtain a heuristic solution for the original ILP. (For example, in the generalised assignment problem, each job must be assigned to exactly one machine. If $\bar{x}(\lambda)$ does not satisfy this condition, one can attempt to convert it into a feasible solution by eliminating multiple assignments and then trying to assign any unassigned jobs to machines having enough free capacity [120].) This can be repeated at each iteration with the corresponding multiplier vector, and one can select the best heuristic solution found.

Fisher [87] called heuristics of this kind "Lagrangian heuristics", but one can view them as a particular kind of matheuristic. They have been applied with great success to several classical combinatorial optimisation problems, such as single machine scheduling [85], the set covering problem [16, 27], the capacitated vehicle routing problem [88], the generalised assignment problem [120], the many-to-many assignment problem [135], and various facility location problems [19, 28, 97, 159].

Lagrangian matheuristics have also been developed for a huge array of more realistic practical problems, such as manpower planning [102], scheduling of energy generators [20, 129], product distribution [29, 183], lot sizing

[38, 182], school timetabling [55], aircraft assignment [67], railway network design [122], fixed-charge problems [191], hybrid flowshop scheduling [146], capacitated network design [116], and the closest string problem in computational biology [179].

We remark that, for a given primal variable $x_j$ and a given multiplier vector $\lambda$, the quantity

$$c_j - \sum_{i=1}^{t} \lambda_i C_{ij}$$

can be viewed as a "Lagrangian reduced cost". These values can be used to guide heuristics, just as we saw for the LP reduced costs in Subsection 2.3 and the "approximate" reduced costs in Subsection 2.4. This idea has been used to particularly good effect in Lagrangian matheuristics for the set covering problem [13, 16, 27, 53, 57] and the capacitated facility location problem [10].

### 3.3 Heuristics based on surrogate relaxation

As far back as 1977, Glover [100] suggested using SR to drive heuristics. As in the case of LR, the idea is to take the solution to the relaxed problem and "repair" it, to make it feasible for the original problem. Although this idea has received less attention than the Lagrangian approach, it has been applied to several problems, such as loading problems [93], manpower planning [102], single-machine scheduling [92], the multidimensional knapsack problem [74, 95, 160], resource-constrained scheduling [71] and a variant of the quadratic knapsack problem [132].

More recently, Dokka *et al.* [73] proposed a general framework for designing matheuristics based on SR. The idea is to exploit the fact that the relaxed problem (6) is usually solved via dynamic programming (DP). The nature of DP is that it constructs and stores a large number of "intermediate" $x$ vectors as it goes along. These $x$ vectors are integral, but unlikely to be feasible for the original ILP. Accordingly, Dokka *et al.* propose to take some or all of the $x$ vectors and "repair" them in the usual way.

Some authors have also experimented with "hybrids" of Lagrangian and surrogate relaxation. See [137], [138] and [96] for applications to the set covering problem, the generalised assignment problem and the maximal covering location problem, respectively.

### 3.4 Variants

We now mention some variations of LR that have also been used to drive heuristics. These variations are presented in more-or-less chronological order.

**Lagrangian dual ascent** *Lagrangian dual ascent* (also known as multiplier adjustment) is a hybrid of LR and dual ascent [86, 110]. It is basically a greedy constructive heuristic for the Lagrangian dual, just as dual ascent is a greedy constructive heuristic for the LP dual. The idea is to set the Lagrangian multipliers to some simple initial values (e.g., zero), and then increase them one at a time, in such a way that the lower bound $L(\lambda)$ is guaranteed to increase monotonically. This approach to solving the Lagrangian dual tends to be much faster than the subgradient method, though it may come at the expense of a weaker lower bound. The method has been used to derive matheuristics for the capacitated vehicle routing problem [89], the generalised assignment problem [90, 111], the uncapacitated facility location problem [108], the segregated storage problem [157], capacity planning in manufacturing [133], the management of cross-docking terminals [155], and various problems in telecommunications [58, 134].

**The restricted Lagrangian approach** The *restricted Lagrangian* approach was introduced by Balas and Christofides [15], in the context of the asymmetric travelling salesman problem (ATSP). Suppose once more that we have an ILP in the form (9), where the problem becomes much easier to solve when the constraints $Cx \geq d$ are dropped. The first step in the approach is to solve the relaxed problem, obtaining a primal vector $x^* \in \mathbb{Z}^n_+$ and a lower bound. After that, we apply a "restricted" form of Lagrangian relaxation, in an attempt to increase the lower bound. We are permitted to assign positive Lagrangian multipliers to one or more of the constraints in the system $Cx \geq d$, but only under the condition that $x^*$ remains optimal for the relaxed problem (or, equivalently, that the Lagrangian reduced cost remains at zero for any variable $x_j$ taking a positive value at $x^*$). Balas and Christofides provide two fast procedures for determining the multipliers.

An interesting feature of the restricted Lagrangian approach is that the set of $x$ variables having zero Lagrangian reduced cost grows during the course of the algorithm. At the end of the procedure, Balas and Christofides use an enumerative procedure to search for an ATSP solution that uses only arcs of zero reduced cost. This last step can be viewed as yet another early example of a matheuristic.

**Additive bounding** A generalisation of the restricted Lagrangian approach, called *additive bounding*, was proposed by Fischetti and Toth [83]. Let us assume for simplicity that our COP has been formulated as an ILP of the form (1). We suppose that there are several fast lower-bounding procedures for our COP, each of which exploits a different substructure of the problem. We also assume that each such procedure returns a dual vector. The procedures are then applied in some chosen sequence. Suppose the first procedure terminates with dual vector $y^1$. We store the corresponding lower bound $b^T y^1$, and replace the original cost vector $c$ with the reduced cost vector $c^1 = c - A^T y^1$. We then feed the reduced cost vector into the second procedure, yielding a new dual vector $y^2$. We then increase our lower bound by $b^T y^2$, and replace the vector $c^1$ with the new vector $c^2 = c^1 - A^T y^2$. This process is repeated until all procedures have terminated.

The additive bounding procedure has been mainly used to fathom nodes within branch-and-bound algorithms. It has however also been used to develop matheuristics. Fischetti and Toth [84] proposed an additive bounding procedure for the ATSP that generates heuristic solutions using an approach similar to the one proposed by Balas and Christofides [15]. Later on, Vigo [186] proposed a heuristic for the asymmetric capacitated vehicle routing problem that makes use of an additive bounding procedure for generating an initial solution. Later still, Caprara [52] used additive bounding to devise an approximation algorithm for the *breakpoint median problem*, a well-known problem in computational biology. We remark that additive bounding was also been used for generating good dual solutions within the Incremental Core approach described at the end of Subsection 2.3.

**Relaxation adaptive memory programming** Relaxation adaptive memory programming (RAMP) was proposed by Rego [167]. It is similar to the hybrid Lagrangian/surrogate approaches mentioned above, but one is permitted to improve both primal and dual solutions along the way, using previously existing metaheuristics (such as scatter search and path relinking [103]). The method has been applied to the capacitated minimum spanning tree problem [168], resource constrained project scheduling [169] and capacitated facility location problems [158].

**Semi-Lagrangian relaxation** *Semi-Lagrangian* relaxation [31] is designed for COPs that have a natural ILP formulation of the form

$$\min \left\{ c^T x : \; Ax = b, \; x \in \mathcal{X} \right\},$$

where $A$, $b$ and $c$ are non-negative and $\mathcal{X}$ is a "reasonably simple" subset of $\mathbb{Z}_+^n$. The idea is that we split the equation system $Ax = b$ into two inequality systems, $Ax \leq b$ and $Ax \geq b$. We then relax the latter in Lagrangian fashion. The relaxed problem takes the form:

$$\min \left\{ c^T x + \lambda^T (b - Ax) : \; Ax \leq b, \; x \in \mathcal{X} \right\}. \tag{7}$$

For some COPs, the relaxed ILP can be solved much more easily than the original. In particular, this happens if variables with a positive objective coefficient in (7) must take the value zero in an optimal solution to (7). In this case, it often happens that a large proportion of the variables can be eliminated.

As usual, one can often repair the solution of the relaxed problem to obtain heuristic solutions for the original COP. Matheuristics of this kind have been developed for the $p$-median problem [31], facility location problems [32, 50, 119, 154], the quadratic assignment problem [192], and the design of multi-commodity distribution centres [193].

# 4 Methods Based on Decomposition

Another important family of matheuristics is composed by those that are based on *Dantzig-Wolfe decomposition* [66] or *Benders decomposition* [34]. We recall the basic ideas of these decomposition schemes in Subsections 4.1 and 4.2. Matheuristics based on the schemes are reviewed in Subsections 4.3 and 4.4. The reader interested in more details is referred to [43, 143, 166].

## 4.1 Recap on Dantzig-Wolfe decomposition

Consider a MIP of the form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x^s \in X^s \quad (s = 1, \ldots, t) \\ & x \in \mathbb{R}^n_+, \end{aligned} \tag{8}$$

where the vector $x$ has been partitioned into sub-vectors $x^1, \ldots, x^s$.

Dantzig-Wolfe decomposition works as follows. For $s = 1, \ldots, t$, let $P^s$ be the convex hull of $X^s$, and let $p^s(1), \ldots, p^s(n_s)$ denote the extreme points of $P^s$. We add a new continuous variable, say $\lambda^s_k$, for $s = 1, \ldots, t$ and $k = 1, \ldots, n_s$. We then reformulate the MIP by replacing the constraints (8) with:

$$\begin{array}{ll} x^s = \sum_{k=1}^{n_s} p^s(k)^T \lambda^s & (s = 1, \ldots, t) \\ \sum_{k=1}^{n_s} \lambda^s_k = 1 & (s = 1, \ldots, t) \\ \lambda^s \in \mathbb{R}^{n_s}_+ & (s = 1, \ldots, t). \end{array}$$

The $x$ variables can then be eliminated if desired.

It can be shown (using a similar argument to the one that Geoffrion [98] used for Lagrangian relaxation) that the LP relaxation of the new MIP gives a lower bound that is at least as strong as the one from the LP relaxation of the original MIP. Moreover, the LP relaxation of the new MIP can be solved by an iterative procedure that starts with a subset of the $\lambda$ variables and uses dual prices to generate other $\lambda$ variables as needed. This procedure is called *column generation*.

If desired, the whole procedure can be embedded within a branch-and-bound framework. This overall approach is called *branch-and-price* [22].

### 4.2 Recap on Benders decomposition

Now consider a MIP of the form

$$\min\Big\{c_1^T x + c_2^T y : \; Ax + By \geq b, \; x \in X, \; y \geq 0\Big\}. \tag{9}$$

We assume for simplicity that this MIP is feasible and bounded.

Benders [34] noted that one can write the MIP as:

$$\min\Big\{c_1^T x + f(x) : \; x \in X\Big\},$$

where $f(x)$ is the optimal solution to the following subproblem:

$$\min\Big\{c_2^T y : \; By \geq b - Ax, \; y \geq 0\Big\}.$$

By LP duality, we have

$$f(x) = \max\left\{(b - Ax)^T w : \; B^T w \leq c_2, \; w \geq 0\right\}. \tag{10}$$

Now, let $p^1, \ldots, p^t$ be the extreme points of the feasible region of (10). By definition, we have

$$f(x) = \max_{1 \leq s \leq t} \left\{(b - Ax)^T p^s\right\}.$$

Thus, the original MIP can be reformulated as:

$$\begin{aligned} \min \quad & c_1^T x + z && (11)\\ \text{s.t.} \quad & z \geq b^T p^s - (B^T p^s)^T x \quad (s = 1, \ldots, t) && (12)\\ & x \in X, && (13) \end{aligned}$$

where (11)-(13) is the *master problem* and constraints (12) are the so-called *Benders' cuts*. Since $t$ is usually huge, the master problem is initially solved with only a small number of Benders' cuts, and the others are generated as needed with an iterative procedure.

### 4.3 Dantzig-Wolfe decomposition heuristics

There are several ways in which one can use Dantzig-Wolfe decomposition within a matheuristic scheme. One way is to use diving heuristics instead of branch-and-price, as we mentioned in Subsection 2.2. Another way is to stop when a desired number of columns has been generated, and then feed the current master MIP into a branch-and-bound solver. An early example of this approach is Agarwal *et al.* [3], who applied it to the capacitated vehicle routing problem.

A third option is to use a fast heuristic for the pricing subproblem, instead of an exact method. One of the first proposals along these lines is presented in [56], where the authors deal with a lot sizing problem with setup times and costs and where columns correspond to suboptimal production plans. It turns out that even the pricing subproblem is $NP$-hard for this problem. Given that optimal solutions to the subproblem are not necessarily needed to produce negative-cost columns, a corridor method (see 5.5) is used to generate them. The overall method is heuristic in nature, given the fact that the pricing subproblem is solved heuristically.

Another work [64], related to lot-sizing, proposes to include in the heuristic column generation scheme outlined above a "fix-and-optimize" procedure (related to relax-and-fix, 2.2), that fixes the non-integer variables produced by the column generation component.

Several other works apply column generation for heuristic search without explicitly mentioning DW decomposition. The addressed range of problems is wide, including technician routing [75], train timetabling [148], electric vehicle routing [76], city logistics [41] and planning in bulk ports [68] among others.

### 4.4 Benders decomposition heuristics

Recently, there has been a surge of interest in heuristics based on Benders decomposition (BD), which is well suited to modelling problems that can be decomposed into two interacting subproblems. Until recently, the possibility of using BD to design efficient heuristics was limited by the need to solve the master problem to optimality in order to obtain at least a valid bound, a task that becomes increasingly complex as newly generated Benders' cuts are added. The increased efficiency of MIP solvers now allows BD to be used as a valuable technique for exploiting problem separability in the design of primal heuristics.

Several real-world applications based on Benders' matheuristics have been described. A first approach attempts to address the inability to produce a feasible solution before the end of cut generation. For example, [124] describes an application to chemotherapy production and delivery, a problem that can be decomposed into two stages, a parallel machine scheduling problem combined with a multi-trip traveling salesman problem. The master problem consists in finding the sequence of trips, and the slave problem is a parallel machine scheduling problem. A solution of the master problem is given to a tabu search heuristic, which derives a heuristic solution.

Another example deals with the identification of the economic operating point of power systems [171]. Here, BD verifies a security criterion through an LP-based master problem, and the subproblem involves the solution of a non-linear program to compute the system energy for transient stability evaluation.

Taking a less problem-dependent approach, [141] describes how the general BD framework of the MIP solver SCIP was extended with two heuristics, a trust region-based heuristic and a large neighborhood search, to improve the BD algorithm and the ability to find primal feasible solutions.

Another notable contribution related to BD was the introduction of the so-called *combinatorial Benders' cuts* [61]. These follow the same structure as classical BD, but the subproblem is an ILP rather than an LP. In case of an infeasible master solution, the subproblem returns combinatorial inequalities to be added as cuts to the master. The method was originally intended for exact MIP solutions, but it has been adapted, possibly more easily than classical BD, to design matheuristics. For example, [11] describes a combinatorial Benders' cut approach to minimizing the number of required toll facilities in a transportation network, and [63] deals with the strip packing problem, where the master problem cuts items into unit-width slices and packs them into the strip, while the subproblem tries to reconstruct the rectangular items by fixing the vertical positions of their unit-width slices.

# 5 Some Other Matheuristics

In this section, we consider some matheuristics that do not necessarily yield dual bounds, yet have proven to be very successful in some applications.

## 5.1 Local branching

*Local Branching* [81] is a method that solve MIPs to perform local search. For brevity, we explain how it works only for pure 0-1 LPs. Let $x^h \in \{0,1\}^n$ be the current incumbent heuristic solution, and let $k$ be a given positive integer parameter. We wish to explore the *k-opt neighbourhood* of $x^h$, by which we mean the set of all feasible solutions that can be obtained by changing the value of no more than $k$ variables. To do this, we define the set $S = \{j \in \{1,\ldots,n\} : x^h_j = 1\}$, and add the following constraint to the 0-1 LP:

$$\sum_{j \notin S} x_j + \sum_{j \in S} (1 - x_j) \leq k.$$

Provided $k$ is small enough, it is likely that the modified 0-1 LP can be solved much more quickly than the original. For this purpose, one can use any decent ILP solver. If the solution to the modified 0-1 LP is cheaper than $x^h$, it takes the place of $x^h$, and the process is repeated.

## 5.2 Very large-scale neighbourhood search

Local branching can be viewed as a special case of *very large-scale neighbourhood* (VLSN) search, which means local search based on neighborhoods whose cardinality is permitted to be huge, and possibly even exponentially large in the number of variables [4, 5].

Broadly speaking, VLSN heuristics can be put into three categories: (i) those that use a heuristic to perform a partial search of the neighborhood;

(ii) those that solve a classic COP (such as a matching problem or network flow problem) to explore the neighborhood fully in polynomial time; and (iii) those that explore the neighbourhood by solving an auxiliary MIP, as we saw above in the case of local branching.

A good example of the first type is the approach in [181], which is based on a one-to-one correspondence between "improving cyclic exchanges" and negative-cost cycles in a certain auxiliary graph. This idea has been applied to vehicle routing problems [181], parallel machine scheduling [94], the graph coloring problem [59] and timetabling problems [150], among others. Other influential approaches of the first type are "large neighborhood search" [175] and "adaptive large neighborhood search" [170].

A good example of the second type of VLSN approach is the "matching" neighborhood for the TSP [173], which was later adapted to several other problem, such as scheduling problems [46, 47] and the generalised assignment problem [152, 153].

As for the third type, we already mentioned local branching above. Another excellent example of the third type is the heuristic for the capacitated vehicle routing problem in [69]. It iteratively removes edges (and short paths) from the incumbent solution, and then solves a small ILP to find the cheapest way to add edges and restore feasibility.

We remark that MIP solvers have become increasingly efficient over the past couple of decades, and now incorporate a variety of advanced techniques to tackle hard problems. Thus, using a MIP solver to explore a neighbourhood is now a perfectly reasonable approach. Using complete MIP models for solving subproblems has been called "MIPping" [82]. The current literature contains several other examples of this. For instance, one can use a MIP solver to solve to optimality a Benders' master problem, in order to obtain a valid lower bound. Figure 3 shows some results that we obtained with such an approach.

To close this subsection, we remark that the "ejection chain", a well-known meta-heuristic proposed by Glover [101] sharing similarities with relax-and-fix (sec. 2.2), with "construct, merge, solve and adapt" (CMSA, [40]) and with "adaptive large neighborhood search" (ALNS, [170]), has also been extended with MP modules ([2, 48]), making it convergent with VLSN and with other similar approaches based on decomposition into subproblems [60].

### 5.3 Dynamic programming heuristics

*Dynamic Programming* (DP), created by Bellman [30], is a classical technique for solving optimisation problems that have a certain "sequential" structure. Although DP was originally intended to be an exact method, it has been used to devise heuristics for a range of COPs, including machine grouping in cellular manufacturing [177], the capacitated minimum

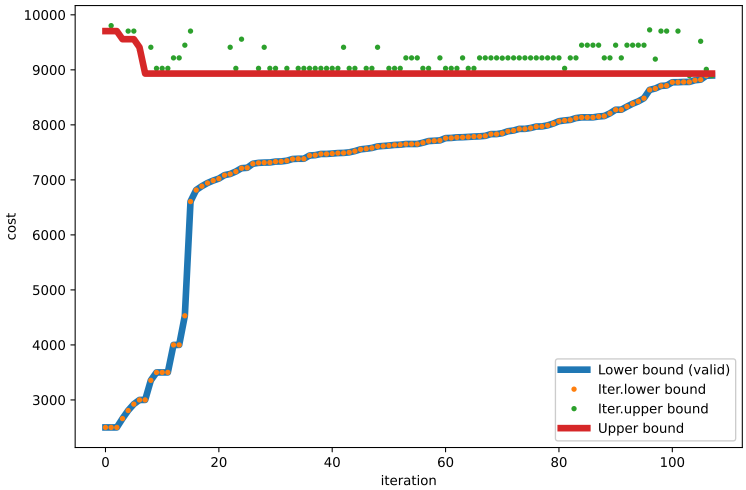


Figure 3: Benders' heuristic, upper and lower bound per iteration

spanning tree problem [104], the TSP with precedence constraints [36], the time-dependent TSP [142], the $p$-median problem [117], the pallet loading problem [174], the multidimensional knapsack problem [35, 149], assembly line balancing [25], and the quadratic knapsack problem [70]. Moreover, as already mentioned in Subsection 3.3, matheuristics have been devised by exploiting the fact that DP is often used when applying surrogate relaxation.

### 5.4 Fore-and-back

*Fore-and-Back* (*FB*) was devised by Bartolini *et al.* [23]. It is suitable for NP-hard problems that have a "natural" DP formulation. (For example, we can formulate the generalised assignment problem (GAP) as a DP, by ordering the jobs and then having one stage for each job, in which we allocate the given job to one of the machines [143].) If FB is given enough computational resources (memory and time), FB is actually an exact method. It was however intended to be a matheuristic, having been designed for quickly finding high quality integer solutions.

FB is an iterated primal-only constructive method, but it computes bounds on the cost of completing partial solutions. This permits it to discard partial solutions from consideration, and sometimes even to compute lower bounds for the entire problem. The basic idea is to perform a series of DP computations, while imposing a restriction on the amount of memory used. The first DP is performed in a "forward" direction, whereas the second is in the "reverse" direction. (In the GAP example, this could correspond to considering jobs in the order $1, 2, \ldots, n$ in the forward phase, but in the order $n, n-1, \ldots, 1$ in the backward phase.)

Following this, a third DP is performed in the forward direction again, and so on until some termination condition is met. Along the way, "promising" partial solutions are stored in memory. In either direction, partial solutions can be tentatively extended to feasible ones using the information stored in the previous DP computation.

FB has been successfully applied to network design problems [24] and generalised assignment problems [143]. Figure 4 shows a trace of an FB run, where at each DP expansion stage we plot the upper and lower bounds computed with the data available at that stage (green and orange dots, respectively). The figure also shows the evolution of the overall best upper and lower bounds as solid lines (red and blue lines, respectively). Note how the lower bounds computed during the forward and backward passes are clearly distinguishable, forming a sort of vertical stripes in the plot. For this particular instance, the first feasible solution is obtained during the first backward pass, and most of the improving feasible solutions are obtained during subsequent backward passes.

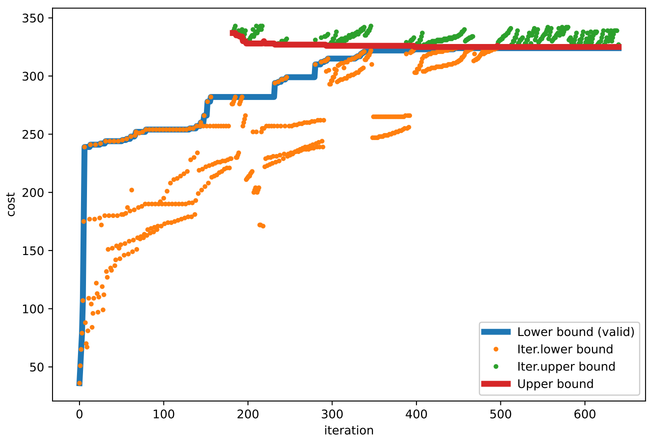


Figure 4: Fore-and-back, upper and lower bound evolution

### 5.5 The corridor method

The *Corridor Method* (CM) [176] is a well-established matheuristic paradigm that originated as a DP adaptation, though it was later extended to other exact approaches such as branch-and-bound. The name comes from the idea of directing the dynamics of expansions in dynamic programming, channelizing them as in a corridor, where the walls are represented by additional constraints imposed on the instance. CM can thus be used to solve problems for which we know an exact method (dynamic programming, also branch-and-bound, branch-and-cut, etc.) that can solve them effectively, but only on smaller instances than the one we are interested in. CM then applies the exact method over successive restricted parts of the solution space, enforcing bounds on the possible expansions of the current state or partial solution, i.e., defining reduced neighborhoods of the current partial solution. Extensions of other search techniques, such as branch-and-bound, may involve searching in the space of feasible solutions, where a first solution is somehow generated and then constraints on its possible adaptations are added. These constraints can be diving like on the absolute number of variables that can be changed, or they can be more problem related, such as the number of facilities that can be opened/closed in facility location problems, or the number of jobs that can be rescheduled in a scheduling problem. In this case, the corridor constraints define local search neighborhoods. A procedure such as that proposed by relax-and-fix (see 2.2) can be seen as working on the linear relaxation of the problem, with the integrality requirements acting as corridor constraints.

Search often results in having to deal with neighborhoods that are exponentially large, but that correspond to instances that can be efficiently solved by the core exact method. The execution is heavily influenced by a control parameter, $\delta_{max}$, which specifies the maximum “width of the corridor”, which is a measure somehow quantifying the maximum size of the subproblems passed to the exact method. It is commonly implemented a simple extension that supports dynamic corridor widths, adjusting the width of the corridor depending on whether or not improving solutions have been found in the current neighborhood. If an improving solution is found in a small neighborhood, the incumbent solution is updated and a new corridor is defined around this new solution. Otherwise, the width of the corridor is increased in the hope of helping to find feasible solutions. Figure 5 shows a trace of a run, showing the iteration upper and lower bounds.

# 6 Discussion

The field of matheuristics has only recently emerged as an independent area of research, but it draws on a rich and extensive body of contributions. In this survey, we have argued that matheuristics offer a key advantage over other meta-heuristics; namely, the use of dual bounds to enable one to terminate early and/or to evaluate the quality of the best-found primal solution. Another nice feature of matheuristics is that they enable one to exploit the relevant progress that has been made in theory, algorithms and software for integer programming.

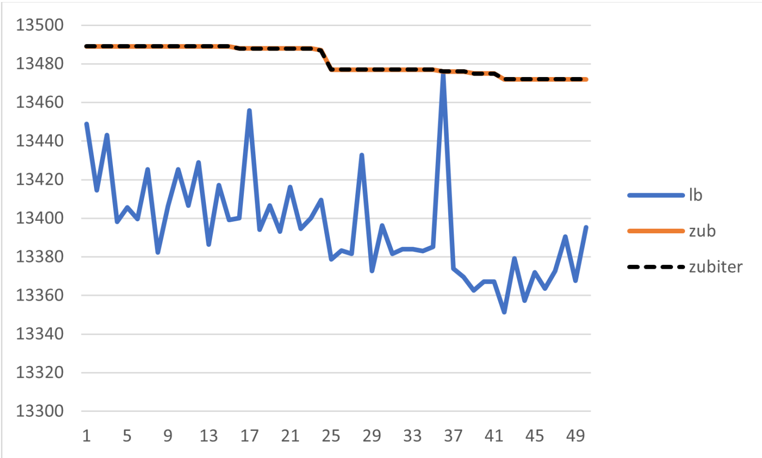


Figure 5: Corridor, upper and lower bound per iteration

Generally speaking, we believe that there is no dominance relation between matheuristics and the more widely used, "mathematics-free" metaheuristics. Nevertheless, relative superiorities can be found on a per-problem basis. In particular, a greater effectiveness of matheuristics has been reported for highly constrained problems, where the feasible solutions are very sparse and thus simple local search becomes expensive and ineffective [143].

We remark that many of the approaches mentioned, such as Lagrangian relaxation, dual ascent and additive bounding, do not rely on the solution of large-scale integer programs at all. For this reason, they are particularly attractive when one is dealing with large-scale problem instances. Moreover, methods such as Lagrangian relaxation and Dantzig-Wolfe decomposition often involve the solution of several independent subproblems, which means that they can exploit parallel processors, and possibly even lead to fully distributed algorithms. Finally, heuristics based on surrogate and semi-Lagrangian relaxation have received relatively little attention so far, and we believe that they merit further study.

# References

[1] R. Aboudi and K. Jörnsten. Tabu search for general zero-one integer programs using the pivot and complement heuristic. *ORSA J. Comput.*, 6:82–93, 1994.

[2] T. Adamo, G. Ghiani, E. Guerriero, and E. Manni. Ejection chain moves for automatic neighborhood synthesis in constrained cardinality-minimization problems. *Int. Trans. Oper. Res.*, 27:1210–1235, 2020.

[3] Y. Agarwal, K. Mathur, and H.M. Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19:731–749, 1989.

[4] R.K. Ahuja, Ö. Ergun, J.B. Orlin, and A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discr. Appl. Math.*, 123:75–102, 2002.

[5] R.K. Ahuja, J.B. Orlin, and D. Sharma. Very large-scale neighborhood search. *Int. Trans. Oper. Res.*, 7:301–317, 2000.

[6] E. Angelelli, R. Mansini, and M.G. Speranza. Kernel search: a general heuristic for the multi-dimensional knapsack problem. *Comput. Oper. Res.*, 37:2017–2026, 2010.

[7] E. Angelelli, R. Mansini, and M.G. Speranza. Kernel search: A new heuristic framework for portfolio selection. *Comput. Optim. Appl.*, 51:345–361, 2012.

[8] C. Archetti, G. Guastaroba, D.L. Huerta-Muñoz, and M.G. Speranza. A kernel search heuristic for the multivehicle inventory routing problem. *Int. Trans. Oper. Res.*, 28:2984–3013, 2021.

[9] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation.* Springer, Heidelberg, 1999.

[10] P. Avella, M. Boccia, A. Sforza, and I. Vasil'ev. An effective heuristic for large-scale capacitated facility location problems. *J. Heuristics*, 15:597–615, 2009.

[11] L. Bai and P.A. Rubin. Combinatorial Benders cuts for the minimum tollbooth problem. *Oper. Res.*, 57:1510–1522, 2009.

[12] A. Balakrishnan, T.L. Magnanti, and R.T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Oper. Res.*, 37:716–740, 1989.

[13] E. Balas and M.C. Carrera. A dynamic subgradient-based branch-and-bound procedure for set covering. *Oper. Res.*, 44:875–890, 1996.

[14] E. Balas, S. Ceria, M. Dawande, F. Margot, and G. Pataki. OCTANE: A new heuristic for pure 0–1 programs. *Oper. Res.*, 49:207–225, 2001.

[15] E. Balas and N. Christofides. A restricted Lagrangean approach to the traveling salesman problem. *Math. Program.*, 21:19–46, 1981.

[16] E. Balas and A. Ho. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. *Math. Program. Study*, 12:37–60, 1980.

[17] E. Balas and C.H. Martin. Pivot and complement—a heuristic for 0-1 programming. *Manag. Sci.*, 26:86–96, 1980.

[18] M.O. Ball. Heuristics based on mathematical programming. *Surv. Oper. Res. Manag. Sci.*, 16:21–38, 2011.

[19] J. Barceló and J. Casanovas. A heuristic Lagrangean algorithm for the capacitated plant location problem. *Eur. J. Oper. Res.*, 15:212–226, 1984.

[20] J.F. Bard. Short-term scheduling of thermal-electric generators using Lagrangian relaxation. *Oper. Res.*, 36:756–766, 1988.

[21] C. Barnhart. Dual-ascent methods for large-scale multicommodity flow problems. *Nav. Res. Logist.*, 40:305–324, 1993.

[22] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: column generation for solving huge integer programs. *Oper. Res.*, 46:316–329, 1998.

[23] E. Bartolini, V. Maniezzo, and A. Mingozzi. An adaptive memory-based approach based on partial enumeration. In V. Maniezzo, R. Battiti, and J.-P. Watson, editors, *Proc. LION '07*, pages 12–24. Springer, Heielberg, 2008.

[24] E. Bartolini and A. Mingozzi. Algorithms for the non-bifurcated network design problem. *J. Heuristics*, 15:259–281, 2009.

[25] J. Bautista and J. Pereira. A dynamic programming based heuristic for the assembly line balancing problem. *Eur. J. Oper. Res.*, 194:787–794, 2009.

[26] J.E. Beasley. An algorithm for set covering problem. *Eur. J. Oper. Res.*, 31:85–93, 1987.

[27] J.E. Beasley. A Lagrangian heuristic for set covering problems. *Nav. Res. Logist.*, 37:151–164, 1990.

[28] J.E. Beasley. Lagrangian heuristics for location problems. *Eur. J. Oper. Res.*, 65:383–399, 1993.

[29] W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Mack, and P.J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13:4–23, 1983.

[30] R.E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[31] C. Beltran-Royo, C. Tadonki, and J.-P. Vial. Solving the $p$-median problem with a semi-Lagrangian relaxation. *Comput. Optim. Appl.*, 35:239–260, 2006.

[32] C. Beltran-Royo, J.-P. Vial, and A. Alonso-Ayuso. Semi-Lagrangian relaxation applied to the uncapacitated facility location problem. *Comput. Optim. Appl.*, 51:387–409, 2012.

[33] G. Belvaux and L.A. Wolsey. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Manag. Sci.*, 46:724–738, 2000.

[34] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[35] D. Bertsimas and R. Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Manag. Sci.*, 48:550–565, 2002.

[36] L. Bianco, A. Mingozzi, S. Ricciardelli, and M. Spadoni. Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming. *INFOR*, 32:19–32, 1994.

[37] O. Bilde and J. Krarup. Sharp lower bounds and efficient algorithms for the simple plant location problem. *Ann. Discr. Math.*, 1:79–97, 1977.

[38] P.J. Billington, J.O. McClain, and L.J. Thomas. Heuristics for multilevel lot-sizing with a bottleneck. *Manag. Sci.*, 32:989–1006, 1986.

[39] R.E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice—closing the gap. In M.J.D. Powell and S. Scholtes, editors, *System Modelling and Optimization*, pages 19–49. Springer, Boston, 2000.

[40] C. Blum, P. Pinacho, López-Ibá nez, and J.A. Lozano. Construct, merge, solve & adapt: a new general algorithm for combinatorial optimization. *Comput. Oper. Res.*, 68:75–88, 2016.

[41] M.A. Boschetti and V. Maniezzo. A set covering based matheuristic for a real-world city logistics problem. *Int. Trans. Oper. Res.*, 22:169–195, 2015.

[42] M.A. Boschetti and V. Maniezzo. Matheuristics: Using mathematics for heuristic design. *4OR*, 20:173–208, 2022.

[43] M.A. Boschetti, V. Maniezzo, and M. Roffilli. Decomposition techniques as metaheuristic frameworks. In V. Maniezzo, T. Stützle, and S. Voß, editors, *Matheuristics*, pages 135–158. Springer, Boston, 2010.

[44] M.A. Boschetti, A. Mingozzi, and S. Ricciardelli. An exact algorithm for the simplified multiple depot crew scheduling problem. *Ann. Oper. Res.*, 127:177–201, 2004.

[45] M.A. Boschetti, A. Mingozzi, and S. Ricciardelli. A dual ascent procedure for the set partitioning problem. *Discr. Optim.*, 5:735–747, 2008.

[46] T. Brueggemann and J.L. Hurink. Two very large-scale neighborhoods for single machine scheduling. *OR Spectrum*, 29:513–533, 2007.

[47] T. Brueggemann and J.L. Hurink. Matching based very large-scale neighborhoods for parallel machine scheduling. *J. Heuristics*, 17:637–658, 2011.

[48] E.K. Burke and T. Curtois. New approaches to nurse rostering benchmark instances. *Eur. J. Oper. Res.*, 237:71–81, 2014.

[49] E. Buson, R. Roberti, and P. Toth. A reduced-cost iterated local search heuristic for the fixed-charge transportation problem. *Oper. Res.*, 62:1095–1106, 2014.

[50] X. Cabezas and S. García. A semi-Lagrangian relaxation heuristic algorithm for the simple plant location problem with order. *J. Oper. Res. Soc.*, pages 1–12, 2022.

[51] V. Cacchiani, C. Contreras-Bolton, L.M. Escobar-Falcón, and P. Toth. A matheuristic algorithm for the pollution and energy minimization traveling salesman problems. *Int. Trans. Oper. Res.*, 30:655–687, 2023.

[52] A. Caprara. Additive bounding, worst-case analysis, and the breakpoint median problem. *SIAM J. Optim.*, 13:508–519, 2002.

[53] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Oper. Res.*, 47:730–743, 1999.

[54] M.E. Captivo. Fast primal and dual heuristics for the $p$-median location problem. *Eur. J. Oper. Res.*, 52:65–74, 1991.

[55] M.W. Carter. A Lagrangian relaxation approach to the classroom assignment problem. *INFOR*, 27:230–246, 1989.

[56] M. Caserta and S. Voß. A math-heuristic Dantzig-Wolfe algorithm for the capacitated lot sizing problem. In Y. Hamadi and M. Schoenauer, editors, *Proc. LION '12*, pages 31–41. Springer, Heidelberg, 2012.

[57] S. Ceria, P. Nobili, and A. Sassano. A Lagrangian-based heuristic for large-scale set covering problems. *Math. Program.*, 81:215–228, 1998.

[58] S.-G. Chang and B. Gavish. Lower bounding procedures for multi-period telecommunications network expansion problems. *Oper. Res.*, 43:43–57, 1995.

[59] M. Chiarandini, I. Dumitrescu, and T. Stützle. Very large-scale neighborhood search: overview and case studies on coloring problems. In C. Blum, M.J. Blesa Aguilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics: An Emerging Approach to Optimization*, pages 117–150. Springer, Heidelberg, 2008.

[60] R. Chirayil Chandrasekharan, P. Smet, and T. Wauters. An automatic constructive matheuristic for the shift minimization personnel task scheduling problem. *J. Heuristics*, 27:205–227, 2021.

[61] G. Codato and M. Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. *Oper. Res.*, 54:756–766, 2006.

[62] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer Programming.* Springer, Cham, 2014.

[63] J.-F. Côté, M. Dell'Amico, and M. Iori. Combinatorial Benders' cuts for the strip packing problem. *Oper. Res.*, 62:643–661, 2014.

[64] J.O. Cunha, H.H. Kramer, and R.A. Melo. Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. *Comput. Oper. Res.*, 104:149–158, 2019.

[65] E. Danna, E. Rothberg, and C. Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Math. Program.*, 102:71–90, 2005.

[66] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Oper. Res.*, 8:101–111, 1960.

[67] M.S. Daskin and N.D. Panayotopoulos. A Lagrangian relaxation approach to assigning aircraft to routes in hub and spoke networks. *Transp. Sci.*, 23:91–99, 1989.

[68] J.L.M. de Andrade and G.C. Menezes. A column generation-based heuristic to solve the integrated planning, scheduling, yard allocation and berth allocation problem in bulk ports. *J. Heuristics*, 29:39–76, 2023.

[69] R. De Franceschi, M. Fischetti, and P. Toth. A new ILP-based refinement heuristic for vehicle routing problems. *Math. Program.*, 105:471–499, 2006.

[70] F. Djeumou Fomeni and A.N. Letchford. A dynamic programming heuristic for the quadratic knapsack problem. *INFORMS J. Comput.*, 26:173–182, 2014.

[71] G. Dobson and I. Khosla. Simultaneous resource scheduling with batching to minimize weighted flow times. *IIE Trans.*, 27:587–598, 1995.

[72] T. Dokka, A.N. Letchford, and M.H. Mansoor. On the complexity of surrogate and group relaxation for integer linear programs. *Oper. Res. Lett.*, 49:530–534, 2021.

[73] T. Dokka, A.N. Letchford, and M.H. Mansoor. Surrogate relaxation as a matheuristic. Technical report, Department of Management Science, Lancaster University, 2021.

[74] T. Dokka, A.N. Letchford, and M.H. Mansoor. Revisiting surrogate relaxation for the multidimensional knapsack problem. *Oper. Res. Lett.*, 50:674–678, 2022.

[75] N. Dupin, R. Parize, and E.-G. Talbi. Matheuristics and column generation for a basic technician routing problem. *Algorithms*, 14:313–352, 2021.

[76] L.C. Echeverri, A. Froger, J.E. Mendoza, and E. Neron. A matheuristic for the multi-period electric vehicle routing problem. In *13th Metaheuristics International Conference*, Cartagena de Indias, Colombia, 2019.

[77] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Oper. Res.*, 26:992–1009, 1978.

[78] B.H. Faaland and F.S. Hillier. Interior path methods for heuristic integer programming procedures. *Oper. Res.*, 27:1069–1087, 1979.

[79] M. Fischetti and M. Fischetti. Matheuristics. In R. Martí, P. Panos, and M.G.C. Resende, editors, *Handbook of Heuristics*, pages 1–33. Springer, Cham, 2016.

[80] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Math. Program.*, 104:91–104, 2005.

[81] M. Fischetti and A. Lodi. Local branching. *Math. Program.*, 98:23–47, 2003.

[82] M. Fischetti, A. Lodi, and D. Salvagnin. Just MIP it! In V. Maniezzo, T. Stützle, and S. Voß, editors, *Matheuristics, Hybridizing Metaheuristics and Mathematical Programming.* Springer, Boston, 2009.

[83] M. Fischetti and P. Toth. An additive bounding procedure for combinatorial optimization problems. *Oper. Res.*, 37:319–328, 1989.

[84] M. Fischetti and P. Toth. An additive bounding procedure for the asymmetric travelling salesman problem. *Math. Program.*, 53:173–197, 1992.

[85] M.L. Fisher. A dual algorithm for the one-machine scheduling problem. *Math. Program.*, 11:229–251, 1976.

[86] M.L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Manag. Sci.*, 27:1–18, 1981.

[87] M.L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15:10–21, 1985.

[88] M.L. Fisher, A.J. Greenfield, R. Jaikumar, and J.T. Lester. A computerized vehicle routing application. *Interfaces*, 12:42–52, 1982.

[89] M.L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11:109–124, 1981.

[90] M.L. Fisher, R. Jaikumar, and L.N. Van Wassenhove. A multiplier adjustment method for the generalized assignment problem. *Manag. Sci.*, 32:1095–1103, 1986.

[91] M.L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Manag. Sci.*, 36:674–688, 1990.

[92] M.L. Fisher, B.J. Lageweg, J.K. Lenstra, and A.H.G. Rinnooy Kan. Surrogate duality relaxation for job shop scheduling. *Discr. Appl. Math.*, 5:65–75, 1983.

[93] J.C. Fisk and M.S. Hung. A heuristic routine for solving large loading problems. *Nav. Res. Logist. Q*, 26:643–650, 1979.

[94] A. Frangioni, E. Necciari, and M.G. Scutellà. A multi-exchange neighborhood for minimum makespan parallel machine scheduling problems. *J. Comb. Optim.*, 8:195–220, 2004.

[95] A. Fréville and G. Plateau. Heuristics and reduction methods for multiple constraints 0–1 linear programming problems. *Eur. J. Oper. Res.*, 24:206–215, 1986.

[96] R.D. Galvão, L.G. Acosta Espejo, and B. Boffey. A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. *Eur. J. Oper. Res.*, 124:377–389, 2000.

[97] R.D. Galvão and L.A. Raggi. A method for solving to optimality uncapacitated location problems. *Ann. Oper. Res.*, 18:225–244, 1989.

[98] A.M. Geoffrion. Lagrangean relaxation for integer programming. *Math. Program. Study*, 2:82–114, 1974.

[99] F. Glover. Surrogate constraint duality in mathematical programming. *Oper. Res.*, 23:434–451, 1975.

[100] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sci.*, 8:156–166, 1977.

[101] F. Glover. Ejection chains and combinatorial leverage for traveling salesman problems. Technical report, School of Business, University of Colorado, Boulder, CO, 1992.

[102] F. Glover, D. Karney, and D. Klingman. A study of alternative relaxation approaches for a manpower planning problem. In Y. Ijiri, A.B. Whinstan, and D. Naddef, editors, *Quantitative Planning and Control*, pages 141–164. Academic Press, New York, 1979.

[103] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control Cybern.*, 29:653–684, 2000.

[104] L. Gouveia and J. Paixao. Dynamic programming based heuristics for the topological design of local access networks. *Ann. Oper. Res.*, 33:305–327, 1991.

[105] H.J. Greenberg and W.P. Pierskalla. Surrogate mathematical programming. *Oper. Res.*, 18:924–939, 1970.

[106] G. Guastaroba, M.W.P. Savelsbergh, and M.G. Speranza. Adaptive kernel search: a heuristic for solving mixed integer linear programs. *Eur. J. Oper. Res.*, 263:789–804, 2017.

[107] G. Guastaroba and M.G. Speranza. Kernel search for the capacitated facility location problem. *J. Heuristics*, 18:877–917, 2012.

[108] M. Guignard. A Lagrangian dual ascent algorithm for simple plant location problems. *Eur. J. Oper. Soc.*, 35:193–200, 1998.

[109] M. Guignard. Lagrangean relaxation. *Trabajos de Operativa (TOP)*, 11:151–228, 2003.

[110] M. Guignard and M.B. Rosenwein. An application-oriented guide for designing Lagrangean dual ascent algorithms. *Eur. J. Oper. Res.*, 43:197–205, 1989.

[111] M. Guignard and M.B. Rosenwein. An improved dual based algorithm for the generalized assignment problem. *Oper. Res.*, 37:658–663, 1989.

[112] P. Hahn and T. Grant. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Oper. Res.*, 46:912–922, 1998.

[113] P. Hansen, J. Brimberg, D. Urošević, and N. Mladenovič. Primal-dual variable neighborhood search for the simple plant-location problem. *INFORMS J. Comput.*, 19:552–564, 2007.

[114] F.S. Hillier. Efficient heuristic procedures for integer linear programming with an interior. *Oper. Res.*, 17:600–637, 1969.

[115] D.S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput.*, 11:555–556, 1982.

[116] K. Holmberg and D. Yuan. A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Oper. Res.*, 48:461–481, 2000.

[117] M. Hribar and M.S. Daskin. A dynamic programming heuristic for the $p$-median problem. *Eur. J. Oper. Res.*, 101:499–508, 1997.

[118] J. Janáček and L. Buzna. An acceleration of Erlenkotter–Körkel's algorithms for the uncapacitated facility location problem. *Ann. Oper. Res.*, 164:97–109, 2008.

[119] K. Jörnsten and A. Klose. An improved Lagrangian relaxation and dual ascent approach to facility location problems. *Comput. Manag. Sci.*, 13:317–348, 2016.

[120] K. Jörnsten and M. Näsberg. A new Lagrangian relaxation approach to the generalized assignment problem. *Eur. J. Oper. Res.*, 27:313–323, 1986.

[121] M.H. Karwan and R.L. Rardin. Surrogate dual multiplier search procedures in integer programming. *Oper. Res.*, 32:52–69, 1984.

[122] M.H. Keaton. Designing optimal railroad operating plans: Lagrangian relaxation and heuristic approaches. *Transp. Res. B: Methodol.*, 23:415–431, 1989.

[123] M.H. Keaton. Designing railroad operating plans: a dual adjustment method for implementing Lagrangian relaxation. *Transp. Sci.*, 26:263–279, 1992.

[124] Y. Kergosien, M. Gendreau, and J.-C. Billaut. A Benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. *Eur. J. Oper. Res.*, 262:287–298, 2017.

[125] S.-L. Kim and S. Kim. Exact algorithm for the surrogate dual of an integer programming problem: subgradient method approach. *J. Optim. Th. Appl.*, 96:363–375, 1998.

[126] J.G. Klincewicz. A dual algorithm for the uncapacitated hub location problem. *Locat. Sci.*, 4:173–184, 1996.

[127] M. Körkel. On the exact solution of large-scale simple plant location problems. *Eur. J. Oper. Res.*, 39:157–173, 1989.

[128] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms.* Springer, Heidelberg, 6th edition, 2018.

[129] G.S. Lauer, N.R. Sandell, D.P. Bertsekas, and T.A. Posbergh. Solution of large-scale optimal unit commitment problems. *IEEE Trans. Power Appar. Syst.*, 1(1):79–86, 1982.

[130] C. Lemaréchal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*, pages 112–156. Springer, Heidelberg, 2001.

[131] A.N. Letchford and S.J. Miller. Fast bounding procedures for large instances of the simple plant location problem. *Comput. Oper. Res.*, 39:985–990, 2012.

[132] L. Létocart, M.-C. Plateau, and G. Plateau. An efficient hybrid heuristic method for the 0-1 exact k-item quadratic knapsack problem. *Pesquisa Operacional*, 34:49–72, 2014.

[133] S.-K. Lim and Y.-D. Kim. Capacity planning for phased implementation of flexible manufacturing systems under budget restrictions. *Eur. J. Oper. Res.*, 104:175–186, 1998.

[134] F.Y.S. Lin and J.R. Yee. A new multiplier adjustment procedure for the distributed computation of routing assignments in virtual circuit data networks. *ORSA J. Comput.*, 4:250–266, 1992.

[135] I. Litvinchev, M. Mata, S. Rangel, and J. Saucedo. Lagrangian heuristic for a class of the generalized assignment problems. *Comput. Math. Appl.*, 60:1115–1123, 2010.

[136] A. Løkketangen and F. Glover. Solving zero-one mixed integer programming problems using tabu search. *Eur. J. Oper. Res.*, 106:624–658, 1998.

[137] L.A.N. Lorena and F.B. Lopes. A surrogate heuristic for set covering problems. *Eur. J. Oper. Res.*, 79:138–150, 1994.

[138] L.A.N. Lorena and M.G. Narciso. Relaxation heuristics for a generalized assignment problem. *Eur. J. Oper. Res.*, 91:600–610, 1996.

[139] J. Maes, J.O. McClain, and L.N. Van Wassenhove. Multilevel capacitated lotsizing complexity and LP-based heuristics. *Eur. J. Oper. Res.*, 53:131–148, 1991.

[140] M.J. Magazine and O. Oguz. A heuristic algorithm for the multidimensional zero-one knapsack problem. *Eur. J. Oper. Res.*, 16:319–326, 1984.

[141] S.J. Maher. Enhancing large neighbourhood search heuristics for Benders' decomposition. *J. Heuristics*, 27:615–648, 2021.

[142] C. Malandraki and R.B. Dial. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *Eur. J. Oper. Res.*, 90:45–55, 1996.

[143] V. Maniezzo, M.A. Boschetti, and T. Stützle. *Matheuristics: Algorithms and Implementations*. Springer, Cham, 2021.

[144] V. Maniezzo, T. Stützle, and S. Voß, editors. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer, Boston, 2010.

[145] R. Mansini and M.G. Speranza. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *Eur. J. Oper. Res.*, 114:219–233, 1999.

[146] K. Mao, Q.-K. Pan, X. Pang, and T. Chai. A novel Lagrangian relaxation approach for a hybrid flowshop scheduling problem in the steelmaking-continuous casting process. *Eur. J. Oper. Res.*, 236:51–60, 2014.

[147] R. Martí, P.M. Pardalos, and M.G. Resende, editors. *Handbook of Heuristics*. Springer, Cham, 2018.

[148] B. Martin-Iradi and S. Ropke. A column-generation-based matheuristic for periodic and symmetric train timetabling with integrated passenger routing. *Eur. J. Oper. Res.*, 297:511–531, 2022.

[149] M. Masmoudi, Y. Adouani, and B. Jarboui. LP relaxation and dynamic programming enhancing VNS for the multiple knapsack problem with setup. *Int. Trans. Oper. Res., to appear.*

[150] C. Meyers and J.B. Orlin. Very large-scale neighborhood search techniques in timetabling problems. In E.K. Burke and H. Rudová, editors, *Proc. PATAT '06*, pages 24–39, Heidelberg, 2007. Springer.

[151] A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco. An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Manag. Sci.*, 44:714–729, 1998.

[152] S. Mitrović-Minić and A.P. Punnen. Very large-scale variable neighborhood search for the generalized assignment problem. *J. Interdiscip. Math.*, 11:653–670, 2008.

[153] S. Mitrović-Minić and A.P. Punnen. Local search intensified: very large-scale variable neighborhood search for the multi-resource generalized assignment problem. *Discr. Optim.*, 6:370–377, 2009.

[154] E. Monabbati. An application of a Lagrangian-type relaxation for the uncapacitated facility location problem. *Jpn. J. Ind. Appl. Math.*, 31:483–499, 2014.

[155] M.F. Monaco and M. Sammarra. A multiplier adjustment algorithm for a truck scheduling and transshipment problem at a cross-docking terminal. *Soft Comput.*, to appear.

[156] I. Murthy. Solving the multiperiod assignment problem with start-up costs using dual ascent. *Nav. Res. Logist.*, 40:325–344, 1993.

[157] A.W. Neebe. An improved, multiplier adjustment procedure for the segregated storage problem. *J. Oper. Res. Soc.*, 38:815–825, 1987.

[158] Ó. Oliveira, T. Matos, and D. Gamboa. A dual RAMP algorithm for single source capacitated facility location problems. *Ann. Math. Artif. Intell.*, 89:815–834, 2021.

[159] H. Pirkul. Efficient algorithms for the capacitated concentrator location problem. *Comput. Oper. Res.*, 14:197–208, 1987.

[160] H. Pirkul. A heuristic solution procedure for the multiconstraint zero-one knapsack problem. *Nav. Res. Logist.*, 34:161–172, 1987.

[161] T. Polzin and S.V. Daneshmand. Improved algorithms for the Steiner problem in networks. *Discr. Appl. Math.*, 112:263–300, 2001.

[162] M. Posta, J.A. Ferland, and P. Michelon. An exact cooperative method for the uncapacitated facility location problem. *Math. Program. Comput.*, 6:199–231, 2014.

[163] J.-Y. Potvin and M. Gendreau. *Handbook of Metaheuristics*. Springer, Cham, 3rd edition, 2019.

[164] P. Raghavan and C.D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[165] G.R. Raidl. Decomposition based hybrid metaheuristics. *Eur. J. Oper. Res.*, 244:66–76, 2015.

[166] G.R. Raidl and J. Puchinger. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In C. Blum, M.J.B. Aguilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics: An Emerging Approach to Optimization*, pages 31–62. Springer, Heidelberg, 2008.

[167] C. Rego. RAMP: A new metaheuristic framework for combinatorial optimization. In R. Sharda, S. Voß, C. Rego, and B. Alidaee, editors, *Metaheuristic Optimization via Memory and Evolution*, pages 441–460. Springer, Boston, 2005.

[168] C. Rego, F. Mathew, and F. Glover. RAMP for the capacitated minimum spanning tree problem. *Ann. Oper. Res.*, 181:661–681, 2010.

[169] R.C.L. Riley and C. Rego. Intensification, diversification, and learning via relaxation adaptive memory programming: a case study on resource constrained project scheduling. *J. Heuristics*, 25:793–807, 2019.

[170] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.*, 40:455–472, 2006.

[171] H. Saberi, T. Amraee, C. Zhang, and Y.D. Zhao. A heuristic Benders-decomposition-based algorithm for transient stability constrained optimal power flow. *Electr. Power Syst. Res.*, 185:106380, 2020.

[172] R. Sadykov, F. Vanderbeck, A. Pessoa, I. Tahiri, and E. Uchoa. Primal heuristics for branch and price: the assets of diving methods. *INFORMS J. Comput.*, 31:251–267, 2019.

[173] V.I. Sarvanov and N.N. Doroshko. Approximate solution of the traveling salesman problem by a local algorithm with scanning neighbourhoods of factorial cardinality in cubic time (in Russian). *Software, Algorithms and Programs*, 31:11–13, 1981.

[174] G. Scheithauer and J. Terno. The G4-heuristic for the pallet loading problem. *J. Oper. Res. Soc.*, 47:511–522, 1996.

[175] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.-F. Puget, editors, *Proc. CP '98*, pages 417–431, Heidelberg, 1998. Springer.

[176] M. Sniedovich and S. Voß. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics*, 35:551–578, 2006.

[177] H.J. Steudel and A. Ballakur. A dynamic programming based heuristic for machine grouping in manufacturing cell formation. *Comput. Ind. Eng.*, 12:215–222, 1987.

[178] T. Stützle and V. Maniezzo. Preface to the special issue on matheuristics and metaheuristics. *Int. Trans. Oper. Res.*, 27:5–8, 2020.

[179] S. Tanaka. A heuristic algorithm based on Lagrangian relaxation for the closest string problem. *Comput. Oper. Res.*, 39:709–717, 2012.

[180] G.M. Thompson. Shift scheduling in services when employees have limited availability: an LP approach. *J. Oper. Manag.*, 9:352–370, 1990.

[181] P.M. Thompson and H.N. Psaraftis. Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Oper. Res.*, 41:935–946, 1993.

[182] W.W. Trigeiro. A dual-cost heuristic for the capacitated lot sizing problem. *IIE Trans.*, 19:67–72, 1987.

[183] T.J. Van Roy and L.F. Gelders. Solving a distribution problem with side constraints. *Eur. J. Oper. Res.*, 6:61–66, 1981.

[184] R.J. Vanderbei. *Linear Programming.* Springer, Cham, 5th edition, 2020.

[185] V.V. Vazirani. *Approximation Algorithms.* Springer, Heidelberg, 2001.

[186] D. Vigo. A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *Eur. J. Oper. Res.*, 89:108–126, 1996.

[187] D. Wedelin. An algorithm for large scale 0–1 integer programming with application to airline crew scheduling. *Ann. Oper. Res.*, 57:283–301, 1995.

[188] D.P Williamson and D.B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, New York, 2011.

[189] L.A. Wolsey. *Integer Programming.* Wiley, Chichester, 1998.

[190] R.T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Math. Program.*, 28:271–287, 1984.

[191] D.D. Wright and C.H. von Lanzenauer. Solving the fixed charge problem with Lagrangian relaxation and cost allocation heuristics. *Eur. J. Oper. Res.*, 42:305–312, 1989.

[192] H. Zhang, C. Beltran-Royo, B. Wang, L. Ma, and Z. Zhang. Solution to the quadratic assignment problem using semi-Lagrangian relaxation. *J. Syst. Eng. Electron.*, 27:1063–1072, 2016.

[193] H. Zhang, C. Beltran-Royo, B. Wang, and Z. Zhang. Two-phase semi-Lagrangian relaxation for solving the uncapacitated distribution centers location problem for B2C E-commerce. *Comput. Optim. Appl.*, 72:827–848, 2019.