Deriving compact extended formulations via LP-based separation techniques

Giuseppe Lancia and Paolo Serafini

Dipartimento di Matematica e Informatica, University of Udine. Via delle Scienze 206, 33100 Udine, Italy.
giuseppe.lancia@uniud.it, paolo.serafini@uniud.it

Abstract. The best formulations for some combinatorial optimization problems are integer linear programming models with an exponential number of rows and/or columns, which are solved incrementally by generating missing rows and columns only when needed. As an alternative to row generation, some exponential formulations can be rewritten in a compact extended form, which have only a polynomial number of constraints and a polynomial, although larger, number of variables. As an alternative to column generation, there are compact extended formulations for the dual problems, which lead to compact equivalent primal formulations, again with only a polynomial number of constraints and variables.
In this this paper we introduce a tool to derive compact extended formulations and survey many combinatorial optimization problems for which it can be applied. The tool is based on the possibility of formulating the separation procedure by an LP model. It can be seen as one further method to generate compact extended formulations besides other tools of geometric and combinatorial nature present in the literature.

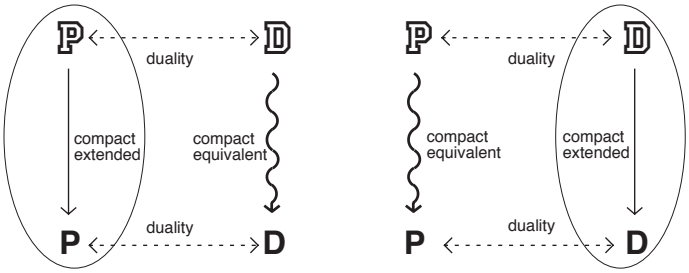
1 Introduction

For some combinatorial optimization problems problems the best ILP formulations, leading to the tightest bounds, employ an exponential number of either constraints or variables. Two famous examples are the TSP, with its exponentially many subtour-inequalities [19] and the Cutting Stock problem with its exponentially many variables [25, 26]. In both cases we deal with a polyhedron described by an exponential number of inequalities. In the former case the polyhedron is the feasible set of the (relaxed) primal ILP formulation whereas in the latter case it is the feasible set of the dual problem.

Both exponential formulations can be adopted in practice since they can be dealt with in an implicit way, provided the "separation" problem (for either the primal or the dual formulation) can be solved in polynomial or pseudopolynomial time. More specifically, if a model has an exponential number of constraints, only a small subset of them is explicitly stored in the LP matrix, and a new row is added to the matrix only if a specific separation procedure (usually called 'pricing' if it refers to the dual problem) finds a missing constraint violated by the current LP solution. The use of separation and pricing within branch-and-bound schemes has lead to the successful *branch-and-cut* [43, 28] and *branch-and-price approaches* [1, 4].

As an alternative to the exponential formulations, it is sometimes possible to describe formulations providing the same tight bounds but employing only a polynomial number of variables and constraints.

In the case of a problem with exponentially many inequalities, these alternative formulations are defined in a higher dimensional space in such a way that the polyhedron of the feasible solutions, once projected back onto the space of the original variables, coincides with the original polyhedron. These new formulations are called *compact extended*, where the adjective "compact" underlines their polynomial-size, while "extended" implies that they are defined over a superset of the set of the original variables. The use of compact extended formulations can provide a direct and often simpler way of solving a problem while exploiting the same strong bounds of the original exponential formulations. We refer the reader to the comprehensive survey on



(a) primal w/exponential n.of rows (b) primal w/exponential n.of cols

Fig. 1. Diagram of derivation of compact formulations

compact extended formulations by [17] and to [33]. The basic tools shown in these surveys are mostly of geometric and combinatorial type, relying on the transformation of the inequalities in the higher dimensional space as they are projected in the original smaller dimensional space. Moreover, there are other tools, like, for instance, the reflection rotations introduced in [34].

In the case of a problem with exponentially many variables (a typical column-generation model), it is clearly the dual problem of the relaxed primal that exhibits exponentially many inequalities. Hence a compact extended formulation can be sometimes defined for the dual problem. This new dual has in turn a dual problem that has a polynomial number of variables and inequalities. This problem is of course in close connection to the primal problem we started with and can be considered its compact reformulation, which we call the *compact equivalent* of the original model. Note that, differently from the compact extended formulation, the compact equivalent formulation cannot be seen as a problem defined in a higher dimensional space. On the contrary, it is defined in a much lower dimensional space than the original one. Furthermore, although the original problem and its compact equivalent solve the same problem, there is no projection operator which allows to identify some variables of the compact version with the original ones. The original ones are exponentially many and cannot find place in any compact reformulation. However, as we will describe by many examples, it is in general not difficult to recover the original solution from the compact one.

In Figure 1 we show the operations that are carried out. The letters P and D refer to the primal and dual problem respectively. In the upper rows (in larger font size) there are the exponential formulations. In the lower rows (smaller size and bold face) there are the compact formulations. Within the ovals there are the problems which have a relation of compact-extended type. A waving arrow underlines a compact equivalent. In Figure 1(a) we illustrate the case of exponentially many rows in the primal. Both primal problems have their dual counterparts that are connected. It is sometimes interesting to investigate this connection which can lead to new insight into the problem. In this case computing the duals is not strictly necessary. In Figure 1(b) we illustrate the case of exponentially many variables in the primal. In this case we have to compute the dual and solve its compact extended model. To retrieve the solution of the original problem we have to compute the dual of the compact extended model.

In this paper we introduce a specific tool to generate compact extended formulations and we apply this tool both to row-generation models and to column-generation models. This tool exploits the possibility of formulating the separation procedure (for either the primal or the dual) by an LP model. We will show how the strong duality property of LP allows rewriting the problem in a compact way. Essentially the approach is similar to that in [17] (for a row generation model): new variables are added to the original variables

thus obtaining an extended formulation, but at the same time the number of inequalities is dropped to a polynomial number, and so we have a compact extended formulation.

Since this approach has been around for a while [39, 12] but it is still not widely known, we aim in this paper to provide a common theoretical background for a variety of problems, taken from many areas of combinatorial optimization, for which the same technique can be used.

We stress that in this survey we are not concerned with establishing, by means of computational experiments, if solving a compact formulation is faster or slower than solving via branch-and-cut/price. From our experience we may say that compact optimization is sometimes faster and sometimes slower than exponential formulations, depending on the problem and the specific instances.

One of the advantages of using a compact extended model is that there is no need for the implementation of separation/pricing routines (in fact, for the implementation of anything at all). The only effort is put into describing the model, which can then be given in input to a standard ILP solver. Another advantage of compact optimization concerns formulations with exponentially many variables. Sometimes, the pricing problem for these formulations becomes difficult at the internal nodes of the branch and bound tree, since fixing variables with branching constraints can destroy the structure of the pricing problem. With compact optimization, this problem is overcome.

As shown also by [17], extended formulations can many times provide new insights and highlight some properties of a problem which were not apparent at first (we will give examples of these insights later on). Among the most important theoretical results on extended formulations we recall the work by [21] showing that there are no compact extended formulations for some known polytopes like, the TSP, the stable set and the cut polytopes.

In this paper we present compact extended and equivalent formulations for several combinatorial optimization problems. Some of these formulations were already known, while some are presented here for the first time. We also extend the concept of compact to pseudo-compact, i.e., formulations in which the number of variables or constraints is pseudo-polynomial.

In Table 1 we provide a summary of the various examples treated in this paper, with a quick reference to the size of the compact formulations and citations to the literature.

The remainder of the paper is as follows. In Section 2 we define the type of problems we are dealing with and we define how, in general terms, a so-called compact model can be derived. In Section 3 we provide a general framework for a particular feature, connected to shortest path problems, which appears in many problems and leads to a network-flow model. In Section 4 we provide a first (or, better said, a zero-th) example of the 'compactification' procedure. This example, although almost useless for the specific problem (which has a polynomial algorithm), is useful to pave the way to the subsequent more complex models. Then, from Section 5 to Section 16 we present the examples listed in Table 1. Some conclusions are drawn in Section 17.

2 Large-scale and compact LP problems

In this paper we deal with large-scale LP problems, i.e., LP problems with an exponential number of either columns or rows. If there are exponentially many columns, then the dual problem has exponentially many rows, so basically the two cases can be dealt with in the same way. As we shall see, a compact extended reformulation replaces the exponentially many inequalities with a polynomial number of inequalities in a higher dimensional space. The number of additional variables must also be polynomial. For a compact extended formulation the feasible polyhedron of the original problem, with an exponential number of facets, is the projection of a higher dimensional polyhedron with a polynomial number of facets. A pseudo-compact

1	2	3	4	5	6	7	8
BIN PACKING	5	C	$O(n\,K)$	$O(n+K)$	p	[25, 26]	[14, 15]
MAX CUT	6	R	$O(n^2)$	$O(n\,m)$	c	[3, 2, 20]	[17, 37]
IND SET	7	R	$O(n^2)$	$O(n\,m)$	c	[16, 24]	[17]
TSP	8	R	$O(n\,m)$	$O(n\,m)$	c	[19, 44]	[12]
MRCT	9	C	$O(n^2\,m)$	$O(n^2\,m)$	c	[22]	[12]
STEINER TREE	10	C	$O(n\,m)$	$O(n\,m)$	c	-	-
BND DEGREE TREE	11	C	$O(n\,m)$	$O(n\,m)$	c	-	-
CYCLE PACKING	12	C	$O(n\,m)$	$O(m^2)$	c	[11]	-
ALT CYCLE PACK	13	C	$O(n^2)$	$O(n^4)$	c	[10]	-
ROBUST KNAPSACK	14	R	$O(n)$	$O(n)$	c	[23]	[7]
JOB SHOP	15	C	$O((\ell+m)\bar{T})$	$O(\ell\,\bar{T})$	p	[36]	[36]
FOLD COMPARISON	16	R	$O(m_1 m_2)$	$O(m_1 n_2 + n_1 m_2)$	c	[35, 9]	[12, 13]

Table 1. 1: Problem name; 2: Section in which the problem is described; 3: C (R) means the columns (rows) are exponentially many in the large-scale formulation; 4: column size of the compact formulation; 5: row size of the compact formulation; 6: c means compact; p means pseudo-compact; 7: literature reference of the large-scale formulation, '-' means there is no previous reference; 8: literature reference of the compact formulation.

reformulation can be described exactly in the same way, with the only difference that "pseudo-polynomial" should replace "polynomial" everywhere.

In this paper we will show several LP exponential models that can be reformulated in a compact or pseudo-compact way. In general, the procedure goes as follows. We first explain in detail the case of column generation. The case of row generation, which has similar features, will be treated at the end of the section. Consider the following large-scale ILP problem:

$$\begin{aligned} \min \ & \sum_{j\in J} c_j\, x_j & & \\ & \sum_{j\in J} A_i^j\, x_j \geq b_i & & i \in I \\ & x_j \geq 0, \text{ integer} & & j \in J \end{aligned} \tag{1}$$

where J is an index set of exponential size, so that the matrix A and the vector c are never given explicitly but they are implicitly defined by some properties of their entries. A convention we use in this paper regards the equation numbering. If we reference an integer linear program as (x), we reference the integrality relaxation of (x) as ($\overline{\text{x}}$). We also use the notation $[n] := \{1, \ldots, n\}$. The dual of ($\overline{1}$) is

$$\begin{aligned} \max \ & \sum_{i\in I} b_i\, u_i & & \\ & \sum_{i\in I} A_i^j\, u_i \leq c_j & & j \in J \\ & u_i \geq 0 & & i \in I. \end{aligned} \tag{2}$$

As is well known, the column-generation scheme for $(\bar{1})$ requires solving the following Master problem

$$\begin{aligned} \min \ & \sum_{j \in \bar{J}} c_j \, x_j \\ & \sum_{j \in \bar{J}} A_i^j \, x_j \geq b_i \qquad i \in I \\ & x_j \geq 0 \qquad\qquad\quad j \in \bar{J} \end{aligned} \tag{3}$$

where $\bar{J} \subset J$ is small and explicitly given. Given optimal dual variables $\bar{u}_i$, $i \in I$, of (3), if they are also feasible in (2), then $\bar{x}_j$, $j \in \bar{J}$, optimal in (3), padded with $\bar{x}_j = 0$, $j \in J \setminus \bar{J}$, is also clearly optimal in $(\bar{1})$, because $\bar{x}$ and $\bar{u}$ satisfy strong duality and are both feasible. Detecting feasibility of $\bar{u}$ in (2) is usually called *pricing*. If $\bar{u}$ is not feasible, then a violated inequality must be found, the corresponding column is added to $\bar{J}$ and the Master problem is solved again. The pricing problem is therefore

$$\min_{j \in J} \ (c_j - \sum_{i \in I} A_i^j \, \bar{u}_i). \tag{4}$$

Of course the pricing problem is never solved by inspection and it is instead solved by another problem that exploits the properties of the matrix A and the vector c. Let us rewrite the pricing problem as

$$\min_{y \in Y} f(y, \bar{u}) \tag{5}$$

Compact and pseudo-compact reformulations of (2) are possible if (5) has a dual problem and strong duality holds. This is always the case if (5) is an LP problem [12, 13]. Let the dual problem of (5) be

$$\max_{z \in Z} g(z, \bar{u}) \tag{6}$$

Then problem (2) can be reformulated as

$$\begin{aligned} \max \ & \sum_{i \in I} b_i \, u_i \\ & \min_{y \in Y} f(y, u) \geq 0 \\ & u \geq 0 \ . \end{aligned} \tag{7}$$

Note that by LP properties $\min_{y \in Y} f(y, u) \leq 0$ since there exists $\bar{u}$ and indices j such that $c_j - \sum_{i \in I} A_i^j \, \bar{u}_i = 0$ and therefore we may equivalently express the condition $\min_{y \in Y} f(y, u) = 0$ as $\min_{y \in Y} f(y, u) \geq 0$. However it is very difficult to express the condition $\min_{y \in Y} f(y, u) \geq 0$ within the constraints on the variable u as in (7). Therefore we exploit strong duality and reformulate (7) as

$$\begin{aligned} \max \ & \sum_{i \in I} b_i \, u_i \\ & \max_{z \in Z} g(z, u) \geq 0 \\ & u \geq 0 \end{aligned} \tag{8}$$

which can be simply rewritten as

$$\begin{aligned} \max \ & \sum_{i \in I} b_i \, u_i \\ & g(z, u) \geq 0 \\ & u \geq 0, \ z \in Z \ . \end{aligned} \tag{9}$$

Therefore the basic idea is to free u to let it adjust itself to a feasibility value in (2) by imposing the feasibility condition $g(z,u) \geq 0$. Although (5) and (6) are just two faces of the same problem, it is only the latter that can be fruitfully used in a compact formulation. If (5) is an LP problem we expect the variables $\bar{u}$ to appear as coefficients in its objective function. Not only (7) is difficult to handle, but we would also have the problem that freeing the $\bar{u}$ variables would yield nonlinear expressions. This difficulty disappears in (9) where the $\bar{u}$ appear as r.h.s. coefficients and can be freed without destroying linearity.

We will only consider pricing problems expressed as LP problems. Hence we may assume that (6) can be expressed for instance as

$$\begin{aligned} \max \ & \sum_{h \in H} \gamma_h \, z_h & & \\ & \sum_{h \in H} \alpha^1_{ih} \, z_h \leq \bar{u}_i & & i \in I \\ & \sum_{h \in H} \alpha^2_{kh} \, z_h \leq \beta_k & & k \in K \\ & z_h \geq 0 & & h \in H, \end{aligned} \tag{10}$$

to obtain a compact reformulation of (2), it is just matter of plugging the condition $\sum_{h \in H} \gamma_h \, z_h \geq 0$ together with the constraints in (10) into (2) in place of the constraints $\sum_{i \in I} A^j_i \, u_i \leq c_j$, $j \in J$, so that (2) becomes

$$\begin{aligned} \max \ & \sum_{i \in I} b_i \, u_i & & \\ & \sum_{h \in H} \gamma_h \, z_h \geq 0 & & \\ & \sum_{h \in H} \alpha^1_{ih} \, z_h \leq u_i & & i \in I \\ & \sum_{h \in H} \alpha^2_{kh} \, z_h \leq \beta_k & & k \in K \\ & u_i \geq 0, z_h \geq 0 & & i \in I, h \in H. \end{aligned} \tag{11}$$

We stress again that, whereas $\bar{u}$ in (10) is constant, u in (11) is variable. We call (11) *the compact reformulation* of (2). Solving (11) is equivalent to solving (2). Since we are interested in the primal variables in ($\overline{1}$) we are interested also in the dual of the compact dual (11). Not surprisingly, this last problem is very close to the original problem and it usually corresponds to a reinterpretation of the original problem ($\overline{1}$), most of the times as a special flow problem. In this paper we will show several examples of this construction. It may also happen that these two problems are actually the same problem. This is particularly true if the large-scale LP corresponds to an original continuous small LP problem. In this case we just get back the original small LP. We show also an example of this situation since it gives an idea of how the general procedure works.

We recall that it is essential that the pricing problem can be formulated as a mathematical programming problem for which strong duality holds. This enables expressing the minimum of the pricing problem as the maximum of its dual and it is this problem which replaces the original constraints.

If the large-scale LP problem has exponentially many rows, like (2), the same technique applies. In this case we need to detect if $\bar{u}$, optimal for a restricted number of inequalities $\bar{J}$, is also feasible for the whole set J. This problem is called *separation* and leads to finding the so called *cutting inequalities*. If the separation problem can be expressed as an LP, then the previous considerations apply in order to replace

the exponentially-many constraints by a compact set of constraints. The resulting LP is a compact extended formulation of the original model.

3 A common feature for path problems

One of the most common features in the examples we are going to present consists in having columns A^j corresponding to paths in an undirected graph $G = (V, E)$. It is convenient to have a common notation for all these problems.

We assume that $V = [n]$ and denote each edge $e \in E$ also by the pair $\{i, j\}$ identifying the edge. If the graph is directed we denote each edge $e \in E$ also by the ordered pair (i, j). Although $\{i, j\}$ is a non-ordered set, we prefer to have a standard representation of the edge by the *ordered* pair $\{i, j\}$ with $i < j$. This convention is useful when we have to implicitly consider a directed version of the graph and we have to replace the edge $\{i, j\}$ with the edges (i, j) and (j, i).

In the sequel, the following computations will appear several times. We find it convenient to present them here as a general framework. It is well-known that a shortest path problem with non-negative lengths δ_e from a source $s \in V$ to a destination $t \in V$ can be expressed as a linear program, namely

$$\begin{array}{ll} \max & y_t - y_s \\ & y_j - y_i \le \delta_e \qquad e = \{i, j\} \in E \\ & y_i - y_j \le \delta_e \qquad e = \{i, j\} \in E. \end{array} \tag{12}$$

Therefore, if we need to constrain all paths between a source s and a destination t to be not shorter than a stated amount ν, this condition can be expressed as

$$\begin{array}{ll} & y_t - y_s \ge \nu \\ & y_j - y_i \le \delta_e \qquad e = \{i, j\} \in E \\ & y_i - y_j \le \delta_e \qquad e = \{i, j\} \in E. \end{array} \tag{13}$$

As we will see later, the constraints (13) will be typically embedded in an LP problem, where both the threshold ν and the lengths δ_e can be the sum of a variable and a constant part, i.e., $\nu = u + f$ and $\delta_e = v_e + g_e$ with u and v_e variables. Then (13) becomes

$$\begin{array}{ll} & y_s - y_t + u \le -f \\ & y_j - y_i - v_e \le g_e \qquad e = \{i, j\} \in E \\ & y_i - y_j - v_e \le g_e \qquad e = \{i, j\} \in E. \end{array} \tag{14}$$

Note that the y variables will appear only in (14) within the LP problem. By taking the dual of this LP, let us denote as ζ the dual variables associated to the constraint $y_s - y_t + u \le -f$ and by ξ^+_{ij} and ξ^-_{ij} the variables associated to the second and third set of constraints respectively. Then, among other constraints, the following constraints (associated to the y variables) are present in the dual:

$$\begin{array}{lll} \displaystyle\sum_{j>s:\{s,j\}\in E} (\xi^+_{sj} - \xi^-_{sj}) - \sum_{j<s:\{j,s\}\in E} (\xi^+_{js} - \xi^-_{js}) = \zeta & & \\ \displaystyle\sum_{j>i:\{i,j\}\in E} (\xi^+_{ij} - \xi^-_{ij}) - \sum_{j<i:\{j,i\}\in E} (\xi^+_{ji} - \xi^-_{ji}) = 0 & & i \in V \setminus \{s, t\} \\ \displaystyle\sum_{j<t:\{j,t\}\in E} (\xi^+_{jt} - \xi^-_{jt}) - \sum_{j>t:\{t,j\}\in E} (\xi^+_{tj} - \xi^-_{tj}) = \zeta & & \\ \xi^+_{ij}, \xi^-_{ij}, \zeta \ge 0 & & \{i, j\} \in E \end{array} \tag{15}$$

The constraints (15) define a flow of value ζ from s to t. On each arc the flow is $\xi_{ij} := \xi^+_{ij} - \xi^-_{ij}$ and obeys flow conservation in each node except the source and the destination. For ease of notation, let us denote as $\Phi(s,t,\zeta)$ the feasible set of flows for (15), so that, instead of explicitly writing down (15) we simply write $\xi \in \Phi(s,t,\zeta)$.

4 Max Flow

In this section we show a very simple example of the general compactification procedure. For this example no new insight is gained for the problem, because we will eventually get exactly the problem we started from. This happens in this case because the exponential formulation of the problem is not combinatorially richer than the usual polynomial formulation.

It is well known that the Max-Flow problem can be formulated in term of paths connecting the source to the sink. Let $\mathcal{P}$ be the set of such paths and $P \in \mathcal{P}$ be a generic path. Let x_P be the flow on the path P. A set of flows is feasible if the sum of flows crossing a particular edge does not exceed the edge capacity. The problem can be formulated as the following model with exponentially many columns:

$$\begin{aligned} \max \quad & \sum_{P\in\mathcal{P}} x_P & & \\ & \sum_{P\in\mathcal{P}:e\in P} x_P \le c_e & e \in E & \\ & x_P \ge 0 & P \in \mathcal{P}. & \end{aligned} \tag{16}$$

The dual of (16) is

$$\begin{aligned} \min \quad & \sum_{e\in E} c_e\, u_e & & \\ & \sum_{e\in P} u_e \ge 1 & P \in \mathcal{P} & \\ & u_e \ge 0 & e \in E. & \end{aligned} \tag{17}$$

Hence dual feasibility amounts to computing shortest paths with edge lengths u_e and requiring that each path must be not shorter than 1. By applying the general framework of the previous section we get the compact version of (17)

$$\begin{aligned} \min \quad & \sum_{e\in E} c_e\, u_e & & \\ & y_s - y_t \le -1 & & \\ & y_j - y_i - u_e \le 0 & e = \{i,j\} \in E & \\ & y_i - y_j - u_e \le 0 & e = \{i,j\} \in E & \\ & u_e \ge 0 & e \in E & \end{aligned} \tag{18}$$

whose dual is turn

$$\begin{aligned} \max \quad & \zeta & \\ & \xi \in \Phi(s,t,\zeta) & \\ & |\xi_e| \le c_e & e \in E \end{aligned}$$

which is exactly the 'normal' formulation of a Max-Flow problem. In this case the exponential and the compact (i.e., the normal) formulations are equivalent in the sense that they have the same optimal value.

This should be no surprise since the Max-Flow problem is a continuous polynomial problem. The question becomes more interesting when a combinatorial problem has a stronger formulation via an exponential model which can embed in the constraint matrix part of the combinatorial structure. In this case a primal compact formulation may be a more useful model than a normal approach based only on binary variables. The examples in the next sections will illustrate this possibility.

The same result can be found by showing that the projection onto the u subspace of the polyhedron defined by (18) is the polyhedron defined by (17). See Theorems 4.6 and 4.7 in [17].

5 Bin Packing

The Bin Packing problem was the first problem to be solved by a column-generation technique [25, 26]. The problem can be formulated as follows. There are n types of items. For each $i \in [n]$, there are m_i items of type i, and each of them has integer size $s_i > 0$. We have to fill all items into bins of integer capacity K by using the minimum number of bins. According to [25, 26] the problem can be formulated as the optimal choice of a set of 'filling patterns'. A filling pattern is a way of filling a bin with some items so that the capacity constraint is satisfied. Essentially, a filling pattern is a feasible solution of an integer knapsack problem with capacity K (with the additional requirement, usually automatically satisfied, that a filling pattern cannot have more than m_i items of type i). Let J be the index set of all filling patterns. We represent the j-th filling pattern by a vector a_i^j, with the meaning that the filling pattern has a_i^j items of type i. By its definition, the vector a_i^j satisfies

$$\sum_{i\in[n]} s_i\, a_i^j \le K.$$

Then the model is the following

$$\begin{aligned} \min\ & \sum_{j\in J} x_j \\ & \sum_{j\in J} a_i^j\, x_j \ge m_i, \qquad i \in [n] \\ & x_j \ge 0 \text{ integer} \qquad j \in J. \end{aligned} \tag{19}$$

The dual of ($\overline{19}$) is

$$\begin{aligned} \max\ & \sum_{i\in[n]} m_i\, u_i \\ & \sum_{i\in[n]} a_i^j\, u_i \le 1, \qquad j \in J \\ & u_i \ge 0 \qquad\qquad i \in [n]. \end{aligned} \tag{20}$$

Pricing a column of ($\overline{19}$), i.e., detecting a violated inequality in (20), is carried out by solving the knapsack problem

$$\begin{aligned} \max\ & \sum_{i\in[n]} \bar{u}_i\, z_i \\ & \sum_{i\in[n]} s_i\, z_i \le K \\ & z_i \ge 0 \text{ integer} \qquad i \in [n] \end{aligned} \tag{21}$$

and checking whether the optimal value of (21) is greater than one. Strong duality does not hold for (21). However, the dynamic-programming recursion of the knapsack problem (21) can be written as the following LP problem of pseudo-polynomial size,

$$\begin{array}{lll} \min & y_K - y_0 & \\ & y_k - y_{k-1} \ge 0, & 1 \le k \le K, \\ & y_k - y_{k-s_i} \ge \bar{u}_i & s_i \le k \le K, \qquad i \in [n] \end{array} \tag{22}$$

so that the maximum in (21) has the same value as the minimum in (22). Hence the constraints $\sum_{i\in[n]} a_i^j \, \bar{u}_i \le 1$, $j \in J$, are equivalent to the constraints in (22) plus the constraint $y_K - y_0 \le 1$. Then (20) can be reformulated as

$$\begin{array}{lll} \max & \displaystyle\sum_{i\in[n]} m_i \, u_i & \\ & y_K - y_0 \le 1 & \\ & y_k - y_{k-1} \ge 0, & 1 \le k \le K, \\ & y_k - y_{k-s_i} - u_i \ge 0, & s_i \le k \le K, \qquad i \in [n], \\ & u_i \ge 0. & \end{array} \tag{23}$$

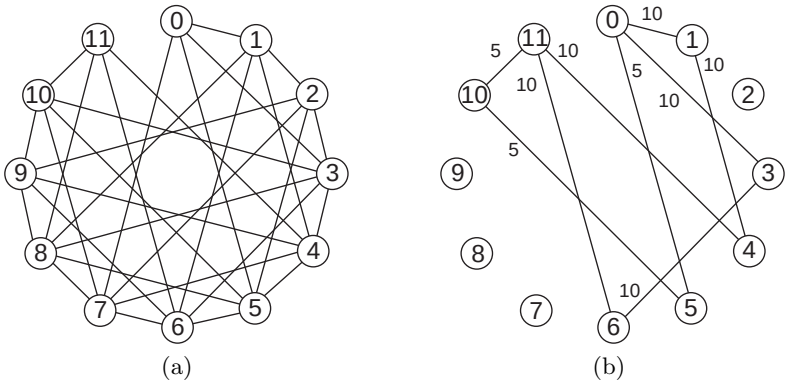
The constraints in (23) can be expressed as constraints on a directed graph $G(V, E)$ where $V = \{0, 1, \ldots, K\}$ and E consists of the arcs $(k-1, k)$, for $k \in [K]$, and $(k - s_i, k)$ for $s_i \le k \le K$, $i \in [n]$. We call the arcs $(k-1, k)$ of type 0 and the arcs $(k - s_i, k)$ of type i. The size of this pseudo-compact formulation is $n + K$ variables and at most $n\,K$ constraints.

The drawback of (23) is that it does not provide a direct information on the actual primal variables, i.e., on how the bins are filled. Moreover, it is the dual of a relaxed formulation and it seems difficult to use (23) in a branch-and-bound search. Therefore, it is convenient to compute the dual of (23). By using the notation introduced in Section 3 the dual is

$$\begin{array}{lll} \min & \zeta & \\ & \xi \in \Phi(0, K, \zeta) & \\ & \displaystyle\sum_{k=s_i}^{K} \xi_{ik} \ge m_i & i \in [n] \\ & \zeta \ge 0, \; \xi_{0k} \ge 0 & k \in [K] \\ & \xi_{ik} \ge 0 & s_i \le k \le K, \qquad i \in [n]. \end{array} \tag{24}$$

The problem (24) is the compact equivalent of (19). It is a special flow problem on G with flow ξ_{0k} on type-0 arcs and flow ξ_{ik} on type-i arcs. There are additional constraints not typical of flow problems because they go across several arcs, i.e., $\sum_{k=s_i}^{K} \xi_{ik} \ge m_i$. The meaning of $\xi_{ik} > 0$ is that one item of type i is put into ξ_{ik} bins increasing the filled quantity in each of these bins from the value $k - s_i$ to the value k. The flows ξ_{0k} correspond to one unit of empty space and enter whenever a bin is not fully filled. A solution of (24) can be decomposed into paths from 0 to K. Every path is a filling pattern, and its flow value corresponds to the number of times the pattern is used. The problem (24) has been proposed by [14] (see also [15]) and its equivalence to the original column generation model has been proved through the Dantzig-Wolfe decomposition technique.

The advantage of this formulation with respect to (23) is that it is possible to impose the integrality requirement directly on the variables ξ.



(a) (b)

Fig. 2. Flow compact models for Bin packing

As an example, let us consider an instance with $n = 6$, $K = 11$, $m = (30, 20, 10)$, $s = (3, 5, 7)$. The graph is shown in Figure 2(a), where all arcs are assumed directed from the lower-indexed node. The solution obtained via (24) is shown in Figure 2(b). In this simple example the solution is integer without imposing integrality and the flow can be uniquely decomposed into three filling patterns: $(1, 0, 1)$ to be used 10 times, $(2, 1, 0)$ 10 times and $(0, 2, 0)$ 5 times, for a total of 25 bins. Note that the total bin space $25 \times 11 = 275$ is equal to $\sum_i m_i\, s_i = 260$ plus the total flow of the arcs of type 0.

6 Max-Cut

Let $G = (V, E)$ be a graph and let w_e, $e \in E$, be positive weights. The Max-Cut problem is defined as the problem of finding the cut of maximum weight. It is well-known [3, 20] that the Max-Cut problem can be modeled as:

$$\begin{aligned} v_A = \max \ & \sum_{e \in E} w_e\, x_e \\ & x(C) \le |C| - 1, \qquad C \in \mathcal{C} \\ & x_e \in \{0, 1\} \qquad\quad e \in E \end{aligned} \tag{25}$$

where $\mathcal{C}$ is the set of odd circuits of G. An equivalent formulation considers the removal of edges so that the resulting graph is bipartite.

$$\begin{aligned} v'_A = \min \ & \sum_{e \in E} w_e\, x_e \\ & x(C) \ge 1, \qquad C \in \mathcal{C} \\ & x_e \in \{0, 1\} \qquad e \in E. \end{aligned} \tag{26}$$

Clearly $v_A + v'_A = \sum_{e \in E} w_e$ also for the relaxed versions. Both problem (25) and (26) can be solved by constraint generation by a simple separation procedure introduced by [29] which can yield a violated inequality. We shall focus on (26). The separation procedure requires defining the graph $\hat{G} = (V' \cup V'', \hat{E})$, where V' and V'' are copies of V and the edge set $\hat{E}$ is

$$\begin{aligned} \hat{E} = \ & \{\{i', j''\} : i' \in V', j'' \in V'', \{i, j\} \in E\} \cup \\ & \{\{i'', j'\} : j' \in V', i'' \in V'', \{i, j\} \in E\}\,. \end{aligned}$$

We may develop a compact extended formulation, as explained in Section 2. Let $y_k(i')$ and $y_k(i'')$ be the optimal dual variables of a shortest path problem in $\hat{G}$ from $k' \in V'$ to $i' \in V'$ and to $i'' \in V''$ respectively. The compact extended formulation of (26) is then (see [37])

$$\begin{array}{lll} \min \ \displaystyle\sum_{e\in E} w_e\, x_e & & \\ y_k(k'') - y_k(k') \geq 1, & & k \in V \\ y_k(i') \ - y_k(j'') \leq x_{\{i,j\}} & & \{i,j\} \in E,\ k \in V \\ y_k(j'') - y_k(i') \ \leq x_{\{i,j\}} & & \{i,j\} \in E,\ k \in V \\ y_k(j') \ - y_k(i'') \leq x_{\{i,j\}} & & \{i,j\} \in E,\ k \in V \\ y_k(i'') \ - y_k(j') \ \leq x_{\{i,j\}} & & \{i,j\} \in E,\ k \in V \\ x_e \in \{0,1\} & & e \in E \end{array} \tag{27}$$

with $4\,n\,m+n$ inequalities and $2\,n^2+m$ variables. The dual of ($\overline{27}$) is a multi-commodity flow problem with flow $\xi^k_{i'j''}$ on $\{i',j''\}$ and $\xi^k_{j'i''}$ on $\{j',i''\}$

$$\begin{array}{lll} \max \ \displaystyle\sum_{k\in V} \zeta^k & & \\ \xi^k \in \Phi(k',k'',\zeta_k) & & k \in V \\ \displaystyle\sum_{k\in V} |\xi^k_{i'j''}| + |\xi^k_{j'i''}| \leq w_e & & e = \{i,j\} \in E. \end{array} \tag{28}$$

This is a particular type of multi-commodity min-cut problem on the graph $\hat{G}$. From each node k' there is a flow ζ^k to k''. The sum of these flows must be maximized, taking into account a special capacity bound which is present jointly on the arcs $\{i',j''\}$ and $\{i'',j'\}$. The problem (28) is the compact equivalent of the dual of ($\overline{26}$), that tries to find a set of odd circulations with maximum total flow within the capacity bounds w_e. The variables ζ^k identify the odd circulations.

We may compare ($\overline{25}$) with the following well-known binary LP model which has four constraints for each possible triple of nodes.

$$\begin{array}{lll} v_B = \max \ \displaystyle\sum_{e\in E} w_e\, z_e & & \\ z_{ij} \leq z_{ik} + z_{jk} & & i<j, k\neq i, k\neq j, i,j,k \in V \\ z_{ij} + z_{ik} + z_{jk} \leq 2 & & i<j<k, i,j,k \in V \\ z_{ij} \in \{0,1\} & & i<j, \end{array} \tag{29}$$

By using projection techniques (see the simple proof in [17]) this model can be also derived from

$$\begin{array}{lll} v_C = \max \ \displaystyle\sum_{e\in E} w_e\, x_e & & \\ x(F) - x(C\setminus F) \leq |F| - 1, & & F \subset C,\ |F| \text{ odd},\ C \in \mathcal{C} \\ x_e \in \{0,1\} & & \end{array} \tag{30}$$

where $\mathcal{C}$ is the set of circuits in G. Not only $v_A = v_B = v_C$, but also the relaxations ($\overline{25}$), ($\overline{29}$) and ($\overline{30}$) have the same value thus providing the same strength in the branch-and-bound search. This result can be also found in [37] and [42].

Basically we have two compact extended formulations for the max cut problem. The formulation (29) has a number of constraints $O(n^3)$ and so it cannot be used on graphs with many nodes. The model (27) has a number of constraints $O(m\,n)$ and so it is a viable alternative if the graph is sparse.

7 Independent Set

Given a graph $G = (V, E)$, and weights w_i for $i \in V$, an *independent* (or *stable*) set is a set $S \subset V$ of nodes no two of which are adjacent, and its weight is defined as $w(S) := \sum_{i \in S} w_i$. The Maximum Independent Set problem calls for determining a maximum-weight independent set, and can be formulated by a binary LP model with variables x_i associated to the nodes of the graph:

$$\begin{array}{ll} \max & \displaystyle\sum_{i \in V} w_i\, x_i \\ & x_i + x_j \le 1 \qquad \{i,j\} \in E \\ & x_i \in \{0,1\} \qquad i \in V. \end{array} \tag{31}$$

This model yields a very weak LP-relaxation bound, but it can be strengthened by the addition of the exponentially many *clique-inequalities*

$$x(K) \le 1 \qquad K \in \mathcal{K} \tag{32}$$

where $\mathcal{K}$ is the set of all cliques in G, and of the exponentially many *odd-cycle inequalities*

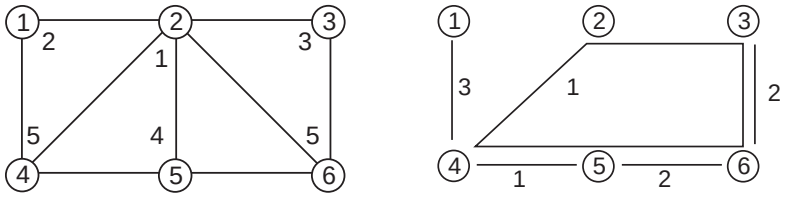
$$x(C) \le \frac{|C|-1}{2} \qquad C \in \mathcal{O} \tag{33}$$

where $\mathcal{O}$ is the set of all odd cycles in G [16]. While constraints (32) have no polynomial-time separation algorithm, constraints (33) can be separated in polynomial time [24] via a reduction to the shortest path problem on the auxiliary graph $\hat{G}$, much in the same way as for odd-cycle inequalities for Max-Cut in Section 6.

Let $y_k(i')$ and $y_k(i'')$ be the optimal dual variables of a shortest path problem in $\hat{G}$ from $k' \in V'$ to $i' \in V'$ and to $i'' \in V''$ respectively. The compact extended formulation of the Independent Set Problem with odd-cycle inequalities is then

$$\begin{array}{lll} \max \displaystyle\sum_{i \in V} w_i\, x_i & & \\ x_i \quad + x_j \quad \le 1, & & \{i,j\} \in E \\ y_k(k'') - y_k(k') \ge 1, & & k \in V \\ y_k(i') \;\; - y_k(j'') \le 1 - x_i - x_j & & \{i,j\} \in E,\ k \in V \\ y_k(j'') - y_k(i') \;\le 1 - x_i - x_j & & \{i,j\} \in E,\ k \in V \\ y_k(j') \;\; - y_k(i'') \le 1 - x_i - x_j & & \{i,j\} \in E,\ k \in V \\ y_k(i'') \;\; - y_k(j') \;\le 1 - x_i - x_j & & \{i,j\} \in E,\ k \in V \\ x_i \in \{0,1\} & & i \in V \end{array} \tag{34}$$

with $4\,n\,m + n + m$ inequalities and $2\,n^2 + n$ variables. This formulation is equivalent, under a transformation of variables, to the one in [17] (Sec. 6.4) obtained by using projection techniques.



(a) An example graph (b) The solution of the Dual

Fig. 3. Independent Set Problem

The dual of $(\overline{34})$ is the compact equivalent of the dual of the exponential formulation with odd-cycle inequalities. It is again a special multi-commodity flow problem on $\hat{G}$ with flow $\xi^k_{i'j''}$ on $\{i', j''\}$ and $\xi^k_{j'i''}$ on $\{j', i''\}$

$$\begin{aligned} \min \sum_{\{i,j\}\in E} \eta_{ij} + \sum_{k\in V} \big(-\zeta^k + \sum_{\{i,j\}\in E} |\xi^k_{i'j''}| + |\xi^k_{j'i''}| \big) & \\ \xi^k \in \Phi(k', k'', \zeta_k) & \qquad k \in V \\ \sum_{\{i,j\}\in\delta(i)} \Big(\eta_{ij} + \sum_{k\in V} |\xi^k_{i'j''}| + |\xi^k_{j'i''}| \Big) \geq w_i & \qquad i \in V. \end{aligned} \tag{35}$$

This problem can be interpreted as follows: in $\hat{G}$ there are flows ζ_k from k' to k''. These flows can be decomposed into paths and circulations. Among the circulations there may be present small circulations $i' \to j'' \to i'$. In addition there are edge variables $\eta_{ij} > 0$. These three quantities may be viewed in G as follows: the paths in $\hat{G}$ are associated to odd cycles in G whereas the small circulations in $\hat{G}$ are associated to edges in G, as are the variables $\eta_{ij} > 0$. Hence the problem (35) is equivalent to find a cover of G by using odd cycles and edges. Each odd cycle has twice the value of the corresponding flow, each edge associated to a small circulation has twice the value of the circulation and each edge associated to $\eta_{ij} > 0$ has value η_{ij}. The cover must be such that for each node the sum of the cycle and edges values must be at least equal to the node weight. The objective consists in finding a minimum cover, taking into account that the cost of an odd cycle C is its value times $(|C| - 1)/2$ and the cost of the edges are equal to their values.

See in Figure 3(a) a graph with weights indicated near the vertices and in Figure 3(b) the dual cover of the graph. The maximum independent set is given by the vertices 4 and 6 for a total weight of 10. The optimal solution of (35) is $\eta_{14} = 3$; $\zeta^1 = 0$ and no flow associated; $\zeta^2 = 0.5$ with flow

$$\xi^2_{2'4''} = 0.5, \quad \xi^2_{4''5'} = 0.5, \quad \xi^2_{5'6''} = 1.5, \quad \xi^2_{6''5'} = 1$$

$$\xi^2_{6''3'} = 1.5, \quad \xi^2_{3'6''} = 1, \quad \xi^2_{3'2''} = 0.5.$$

This flow is decomposed into the path $2' \to 4'' \to 5' \to 6'' \to 3' \to 2''$ with value 0.5 and the two small circulations $5' \to 6'' \to 5'$ of value 1 and $3' \to 6'' \to 3'$ of value 1. The path is associated to the odd cycle $(2, 4, 5, 6, 3)$ in G of value 1 and the small circulations are associated to the edges $(5, 6)$ and $(3, 6)$ in G both with value 2; $\zeta^3 = 0$ with no flow associated; $\zeta^4 = 0$ but there is small circulation $\xi^4_{4'5''} = \xi^4_{5''4'} = 0.5$ which is associated to the edge $(4, 5)$ with value 1; $\zeta^5 = \zeta^6 = 0$ with no flows associated.

The constraints at the vertices are satisfied and the objective value is

$$3 + 0.5 \cdot 4 + 2 + 2 + 1 = 10$$

8 TSP - Subtour inequalities

The (symmetric) traveling salesman problem (TSP) is perhaps the most famous combinatorial problem in the literature. An undirected graph $G = (V, E)$ with nonnegative lengths d_e on the edges is given. A *hamiltonian tour* is a cycle visiting each node in V exactly once. We want to determine the hamiltonian tour of smallest length.

The standard IP formulation for the TSP problem is based on binary variables x_e for each edge $e \in E$. There are *degree constraints* $x(\delta(\{i\})) = 2$, for each $i \in V$, that force each node to be entered and exited exactly once by a feasible solution. Furthermore, in order to avoid subtours from a solution [19], there are $\Omega(2^n)$ *subtour inequalities* $x(\delta(S)) \geq 2$, for each $S \subset V$, where $\delta(S)$ is the set of edges in the cut induced by S.

The standard way of separating the subtour inequalities is via a min-cut problem, which can be solved polynomially [44]. A slightly slower separation procedure is to solve $(n-1)$ maximum flow problems (see, e.g., [18]). Namely, for a fractional solution x^*, each edge e is assigned a capacity interval $[-x^*_e, x^*_e]$. Since it is enough to consider only sets $S \neq V$ such that $1 \in S$, there are no violated constraints if and only if the maximum flow from 1 to each other vertex in V has value at least 2.

Define variables ξ^k_{ij} to represent the flow, possibly negative, on the arc $\{i, j\}$ corresponding to the max-flow problem from 1 to k, for all $k \in V \setminus \{1\}$ and $\{i, j\} \in E$. Let us assume conventionally that a flow from i to j with $i < j$ is positive. We have then the following compact extended formulation of the TSP where the max-flow constraints replace the subtour inequalities [12]:

$$\begin{array}{lll} \min & \displaystyle\sum_{e \in E} d_e \, x_e & \\ & \displaystyle\sum_{e \in \delta(\{i\})} x_e \; = 2 & i \in V \\ & \displaystyle\sum_{j} \xi^k_{1j} \geq 2 & k \in V \setminus \{1\} \\ & \displaystyle\sum_{j>i} \xi^k_{ij} = \sum_{j<i} \xi^k_{ji} & k \in V \setminus \{1\},\ i \in V \setminus \{1, k\} \\ & -x_{ij} \leq \xi^k_{ij} \leq x_{ij} & k \in V \setminus \{1\}, \{i, j\} \in E \\ & x_e \in \{0, 1\} & e \in E. \end{array} \tag{36}$$

In (36) there are $m\,n$ variables and $n^2 - n + 1 + 2\,m\,(n-1)$ constraints.

9 Minimum routing cost trees

The Minimum Routing Cost Tree (MRCT) is the following network–design problem [32, 45, 22]. We are given an undirected weighted graph $G = (V, E)$ in which the length of an edge $e = \{i, j\}$ is denoted as d_e. A *pair* of vertices is an edge of the complete graph $K_n = (V, Q)$. For a spanning tree T and a pair $\{i, j\} \in Q$ of vertices, $d(i, j, T)$ is the length of the unique path connecting i and j in T. The routing cost of T is defined as $r(T) := \sum_{\{i,j\} \in Q} d(i, j, T)$. We want to determine a spanning tree of minimum routing cost.

For each pair $q = \{i, j\} \in Q$, we denote by $\mathcal{P}^q$ the set of simple paths in G between i and j. Conventionally, for each pair $\{i, j\} \in Q$, a path starts in $i < j$. Let $\mathcal{P} = \cup_{q \in Q} \mathcal{P}^q$. For each path $P \in \mathcal{P}$, we let $d_P := \sum_{e \in P} d_e$.

The MRCT problem is formulated [22] as an integer program with decision variables z_P, $P \in \mathcal{P}$, used to select a path between each pair of vertices and x_e, $e \in E$, used to select the tree edges. The constraints are such that, in a feasible solution, the set $\{e \in E \,|\, x_e = 1\}$ defines a tree. The ILP model is the following:

$$\begin{array}{lll} \min & \displaystyle\sum_{q\in Q}\sum_{P\in\mathcal{P}^q} d_P\, z_P & \\ & \displaystyle\sum_{P\in\mathcal{P}^q} z_P \geq 1 & q \in Q \\ & \displaystyle -\sum_{P\in\mathcal{P}^q : e\in P} z_P + x_e \geq 0 & e \in E, q \in Q \\ & \displaystyle\sum_{e\in E} x_e = n-1 & \\ & z_P \geq 0, x_e \geq 0 \text{ integer.} & \end{array} \tag{37}$$

Let u_q, v_{eq} and w be the dual variables associated to the three groups of constraints in ($\overline{37}$). The columns to be generated are those corresponding to the variables x_P and are associated to violated dual constraints $u_q - \sum_{e\in P} v_{eq} \leq d_P$, that can be rewritten as $\sum_{e\in P}(v_{eq} + d_e) \geq u_q$. Hence a pricing algorithm [22], consists in finding, for each pair $q = \{i, j\}$, the shortest i–j path in E, with respect to the costs $(v_{eq} + d_e)$, and checking if it is shorter than u_q.

Let y^q_k, for $k \in V$ and $q \in Q$, represent the length of the shortest i–k path. The compact extended formulation of the dual of ($\overline{37}$) (see [12]) is

$$\begin{array}{lllll} \max & \displaystyle\sum_{q\in Q} u_q + (n-1)\, w & & & \\ & u_q & - y^q_j + y^q_i \leq 0 & q = \{i,j\} \in Q, \quad i < j \\ & - v_{eq} & + y^q_h - y^q_k \leq d_e & q \in Q, \quad e = \{h,k\} \in E \\ & - v_{eq} & + y^q_k - y^q_h \leq d_e & q \in Q, \quad e = \{h,k\} \in E \\ & \displaystyle\sum_{q\in Q} v_{eq} + w & \leq 0 & e \in E & \\ & u_q \geq 0,\ v_{eq} \geq 0. & & & \end{array} \tag{38}$$

The size of the compact extended formulation is $n\,(n{-}1)\,(m{+}1)/2{+}1$ variables and $n\,(n{-}1)\,(2\,m{+}1)/2{+}m$ constraints. This compact dual, however, does not provide a direct information on the routing tree. To this aim, we compute the dual of (38), which is the following compact equivalent of ($\overline{37}$)

$$\begin{array}{lll} \min & \displaystyle\sum_{\{h,k\}\in E} d_{hk} \sum_{q\in Q} |\xi^q_{hk}| & \\ & \xi^q \in \Phi(i,j,1) & q = \{i,j\} \in Q \\ & |\xi^q_{hk}| \leq x_e & q \in Q, \{h,k\} = e \in E \\ & \displaystyle\sum_{e\in E} x_e = n-1. & \end{array} \tag{39}$$

This model turns out to be a min-cost multi-commodity flow problem with an additional constraint. For each pair $\{i, j\}$ one unit of flow must go from i to j. On every arc e, a capacity x_e is available for each flow.

Furthermore, the total capacity in the network is limited and must be equal to $n-1$. In an integral solution the value of x_e is either 1 or 0 and clearly the objective measures the routing cost.

10 Steiner trees

The Steiner Tree Problem can be stated as follows: given a graph $G=(V,E)$, $V=[n]$, costs $c_e>0$ for each arc $e\in E$, and a subset $T\subset V$ of *terminal nodes*, find a minimum-cost tree spanning all nodes in T. The nodes $V\setminus T$ may or may not be spanned by the tree. Nodes in $V\setminus T$ spanned by an optimal tree are called *Steiner nodes*.

For notational simplicity, let us suppose $1\in T$. Let $\mathcal{P}_j$ be the set of paths from 1 to j with $j\in T$. Let $T'=T\setminus 1$. We model the problem as in the previous section by introducing decision variables z_P, used to select a path between 1 and $j\in T$ and x_e, $e\in E$, used to select the tree edges. The constraints are such that, in a feasible solution, the set $\{e\in E\,|\,x_e=1\}$ defines a tree.

$$\begin{array}{lll}
\min & \displaystyle\sum_{e\in E} c_e\,x_e & \\
& \displaystyle\sum_{P\in\mathcal{P}_j:e\in P} z_P\le x_e & j\in T',\quad e\in E\\
& \displaystyle\sum_{P\in\mathcal{P}_j} z_P\ge 1 & j\in T'\\
& x_e\ge 0,\ \text{integer} & e\in E\\
& z_P\ge 0 & P\in\mathcal{P}_j,\quad j\in T'.
\end{array}\tag{40}$$

Note that an optimal solution of (40) is necessarily a tree and there is no need to impose the constraint $\sum_{e\in E}x_e\le n-1$. Let v_j^e and u_j be the dual variables associated, respectively, to the first and second set of constraints in ($\overline{40}$). The pricing problem consists in solving shortest path problems for each $j\in T'$ with arc lengths v_j^e. By writing the conditions for an optimal path, the dual compact extended model is

$$\begin{array}{lll}
\max & \displaystyle\sum_{j\in T'} u_j & \\
& y_j^j-y_1^j\ge u_j & j\in T'\\
& y_h^j-y_k^j\le v_j^{hk} & j\in T',(h,k)\in E\\
& y_k^j-y_h^j\le v_j^{hk} & j\in T',(h,k)\in E\\
& \displaystyle\sum_{j\in T'} v_j^e\le c_e & e\in E\\
& v_j^e,u_j\ge 0 & P\in\mathcal{P}_j,\quad j\in T'
\end{array}\tag{41}$$

with y_h^j being the shortest distance from 1 to h with lengths v_j^e. The model (41) has $|T'|\,(n+m+1)+1=O(n\,m)$ variables (by assuming that $|T'|$ is linearly related to n) and $|T'|\,(2\,m+1)+m=O(n\,m)$ constraints.

The dual of (41), i.e., the compact equivalent of ($\overline{40}$), is

$$\begin{array}{lll} \min & \displaystyle\sum_e c_e\, x_e & \\ & \xi^j \in \Phi(1,j,1), & j \in T' \\ & |\xi^j_e| \le x_e & j \in T',\ e \in E. \end{array} \tag{42}$$

The intepretation of this flow problem is similar as for MRCT problem. A unit of flow must be sent from one terminal node to all other terminal nodes, thus assuring connectivity. A capacity (independent of each flow) is available on each arc. Its use implies a cost and we want to minimize the total cost.

In case $T = V$ the Steiner tree problem becomes the well known Minimal Spanning Tree problem, which is polynomial. The convex hull of the incidence vectors of spanning trees is a polyhedron with facets given by $x(S) \le |S| - 1$, for each $S \subset V$ plus the equality $x(V) = |V| - 1$. Hence, this is an exponential formulation which admits the compact extended formulation (42), which is the same as the one in [17] Sec. 4.3 when $T = V$.

11 Bounded-degree spanning trees

The Bounded-degree Minimum Spanning Tree problem consists in finding a minimum-cost spanning tree with node degrees bounded by a given number. A similar problem consists in finding a spanning tree whose maximum node degree is minimum (among all spanning trees). We show the exponential and compact models for the second problem only since there is no difficulty in doing the same for the first problem.

As in Section 10, let $\mathcal{P}_j$ be the set of paths from 1 to j with $j \in V \setminus 1$, z_P be decision variables to select a path $P \in \mathcal{P}_j$ between 1 and j and x_e be variables selecting the edges $e \in E$ of the tree. Let r be a variable denoting the maximum degree in the spanning tree. The ILP model for this problem is similar to (40) with the following differences: V replaces T, the objective function is r and, in addition, there are inequalities $\sum_{e \in \delta(i)} x_e \le r$, $i \in V$ with corresponding dual variables w_i. Also in this case the pricing problem consists in solving shortest path problems for each $j \in V \setminus 1$ with arc lengths v^e_j. By exploiting the conditions for an optimal path, the dual compact extended model is

$$\begin{array}{lll} \max & \displaystyle\sum_{j \in V \setminus 1} u_j & \\ & y^j_j - y^j_1 \ge u_j & j \in V \setminus 1 \\ & y^j_h - y^j_k \le v^{hk}_j & j \in V \setminus 1, \{h,k\} \in E \\ & y^j_k - y^j_h \le v^{hk}_j & j \in V \setminus 1, \{h,k\} \in E \\ & \displaystyle\sum_{j \in V \setminus 1} v^e_j \le w_h + w_k & e = \{h,k\} \in E \\ & \displaystyle\sum_{i \in V} w_i = 1 & \\ & v^e_j, u_j, w_i \ge 0 & \end{array} \tag{43}$$

with y^j_h shortest distance from 1 to h with lengths v^e_j. The model (43) has $(n-1)(n+m+1)+1 = O(n\,m)$ variables and $(n-1)(2\,m+1)+m = O(n\,m)$ constraints. The dual of (43), i.e., the compact equivalent of

the original problem, is

$$\begin{aligned}
\min\ & r \\
& \xi^j \in \Phi(1,j,1), && j \in V \setminus 1 \\
& |\xi^j_e| \le x_e && j \in V \setminus 1,\ e \in E \\
& \sum_{e \in \delta(i)} x_e \le r.
\end{aligned}$$

The intepretation of this flow problem is very close to that for the Steiner tree problem. A unit of flow must be sent from one node to all other nodes, thus assuring connectivity. On each arc there is available a capacity which induces a node capacity as the sum of the capacities of all incident arcs. The largest node capacity has to be minimized.

12 Cycle packing

A well-known optimization problem consists in finding a cycle packing of maximum cardinality in a graph $G = (V, E)$. There exists both a directed and an undirected version of this problem. The problem can be naturally modeled as the following large-scale binary LP problem [11]

$$\begin{aligned}
\max\ & \sum_{C \in \mathcal{C}} x_C \\
& \sum_{C \in \mathcal{C}: e \in C} x_C \le 1 && e \in E \\
& x_C \in \{0,1\} && C \in \mathcal{C}
\end{aligned} \tag{44}$$

with $\mathcal{C}$ the set of cycles in G. The pricing can be carried out by finding a cycle of minimum weight, with weights given by the optimal duals $\bar{u}_e$ in $(\overline{44})$. We consider here the undirected case. We leave to the reader the directed case which can be approached via a bipartite matching problem much in the same style as the problem we will face in the next section.

A minimum cycle in an undirected graph can be found as follows. First compute, for each $\{h,k\} \in E$, a shortest path P^{hk}, not containing the edge $\{h,k\}$, between h and k. Then form the cycles $C^{hk} := \{h,k\} \cup P^{hk}$ and take the minimum among these cycles. For the minimum cycle to be not shorter than 1, we can impose constraints involving variables y^{hk}_i, $\{h,k\} \in E$, $i \in V$, that lead to the following compact extended of the dual of $(\overline{44})$

$$\begin{aligned}
\min\ & \sum_{e \in E} u_e \\
& y^{hk}_h - y^{hk}_k + u_{hk} \ge 1 && \{h,k\} \in E \\
& y^{hk}_i - y^{hk}_j + u_{ij} \ge 0 && \{h,k\} \in E,\ \{i,j\} \in E \setminus \{h,k\} \\
& y^{hk}_j - y^{hk}_i + u_{ij} \ge 0 && \{h,k\} \in E,\ \{i,j\} \in E \setminus \{h,k\} \\
& u_e \ge 0 && e \in E.
\end{aligned} \tag{45}$$

The size of this formulation is $n\,m + m$ variables and $m\,(2m-1)$ constraints. The dual of (45), i.e., the compact equivalent of ($\overline{44}$), is

$$\begin{aligned} \max \quad & \sum_{\{h,k\}\in E} \zeta^{hk} \\ & \xi^{hk} \in \Phi(k,h,\zeta^{hk}) && \{h,k\}\in E \\ & \zeta^{hk} + \sum_{\{i,j\}\in E\setminus\{h,k\}} |\xi^{ij}_{hk}| \le 1 && \{h,k\}\in E \end{aligned}$$

The interpretation of this flow problem is as follows: for each edge $\{h,k\}$ a flow ζ^{hk} is starting from one endpoint (say h) moving on the network, without using the edge $\{h,k\}$, and ending in k. This flow is actually a circulation of value ζ^{hk} passing through the edge $\{h,k\}$. On each edge the sum of all circulations must not exceed 1.

13 Alternating cycle decomposition problem

In this section we present a problem derived from a computational biology application [10]. A graph $G = (V,E)$ is given with arcs partitioned into two classes which we may call 'colors', say black and white. The graph has the property that, in each node, the number of black incident arcs is equal to the number of white incident arcs. An alternating cycle is a cycle made of edges alternating the two colors and not strictly containing a cycle that alternates the two colors. If there are no edge repetitions, the alternating cycle is called *genuine*, and we denote by $\mathcal{C}_0$ the set of genuine alternating cycles. If there are edge repetitions the alternating cycle is called *spurious*, and we denote by $\mathcal{C}_1$ this set of cycles. Let $\mathcal{C} := \mathcal{C}_0 \cup \mathcal{C}_1$. An alternating cycle decomposition is a set of disjoint genuine alternating cycles, i.e., each edge is present in at most one alternating cycle. The alternating cycle decomposition problem consists in finding a decomposition of maximum cardinality. This problem is NP-hard [8]. A natural set packing 0-1 LP model is the following

$$\begin{aligned} \max \quad & \sum_{C\in\mathcal{C}_0} x_C \\ & \sum_{C\in\mathcal{C}_0: e\in C} x_C \le 1 && e\in E \\ & x_C \in \{0,1\} && C\in\mathcal{C}_0. \end{aligned} \tag{46}$$

Pricing the columns can be carried out by solving a perfect matching problem on an auxiliary non-bipartite graph [10]. In principle, the theory presented in this paper could be applied to this particular pricing since matching problems do have dual problems for which strong duality holds. However, LP formulations of matching problems require an exponential number of inequalities and this rules out the possibility of a compact formulation.

In [10] a weaker LP model is presented for the alternating cycle decomposition problem which calls for a simpler pricing problem. The weaker set packing 0-1 LP model is the following

$$\begin{aligned} \max \quad & \sum_{C\in\mathcal{C}} x_C \\ & \sum_{C\in\mathcal{C}} \mu_{e,C}\, x_C \le 1 && e\in E \\ & x_C \in \{0,1\} && C\in\mathcal{C} \end{aligned} \tag{47}$$

where $\mu_{e,C}$ is equal to the number of times the arc e is traversed by the cycle C. We note that the 0-1 linear programs (46) and (47) are equivalent since only genuine alternating cycles can have $x_C = 1$ in (47). However, the linear relaxations ($\overline{46}$) and ($\overline{47}$) are different. The possibility of fractional values for x_C allows for spurious cycles to be present in the solution of ($\overline{47}$). The dual of ($\overline{47}$) is

$$\begin{aligned} \min \ & \sum_{e\in E} u_e & & \\ & \sum_{e\in E} \mu_{e,C}\, u_e \geq 1 & & C \in \mathcal{C} \\ & u_e \geq 0 & & e \in E. \end{aligned} \tag{48}$$

Hence, pricing calls for finding an alternating cycle of minimum weight with weights given by the optimal dual variables $\bar{u}_e$ of the master problem. A minimum alternating cycle can be found by the following construction [10]. An auxiliary bipartite graph $\hat{G} = (V' \cup V'', E' \cup E'')$ is defined where V' and V'' are copies of V. An arc $e' = \{i', j''\} \in E'$, $i' \in V'$, $j'' \in V''$, exists in $\hat{G}$ if there exists a node $k \in G$ such that $\{i, k\} \in E$ is white and $\{k, j\} \in E$ is black. Let us denote this set of intermediate nodes as $K(i, j)$. These edges receive weights $\min_{k\in K(i,j)} \bar{u}_{ik} + \bar{u}_{kj}$. Moreover, there exist edges $\{i', i''\} \in E''$ for each $i \in V$. These edges receive zero weight. If we exclude the perfect matching E'', let us call it 'trivial matching', each perfect matching in $\hat{G}$ corresponds to one or more alternating cycles in $\mathcal{C}$ and a minimum alternating cycle in $\mathcal{C}$ corresponds to a minimum perfect matching. Since the trivial matching has zero weight, it is indeed the absolute minimum matching. Hence, if we want to find a minimum alternating cycle we have to explicitly exclude the trivial matching. This is straightforward if we solve the matching problem via ILP by adding the constraint

$$\sum_{i\in V} z(i', i'') \leq n - 1$$

to the usual LP formulation of a bipartite matching problem with variables $z(i', j'') \in \{0, 1\}$ for each $\{i', j''\} \in E' \cup E''$. However, adding this constraint destroys the total unimodularity property of the bipartite matching constraint matrix, so that strong duality is lost.

Hence we have to resort to a more complex construction in which we have to carry out n pricing problems by excluding from $\hat{G}$ each arc $\{i', i''\}$ in turn. Let E''_h be the edge set E'' with the edge $\{h', h''\}$ removed. Each pricing problem is therefore solved by the following LP problem

$$\begin{aligned} \min \ & \sum_{\{i',j''\}\in E'} \min_{k\in K(i,j)} (\bar{u}_{ik} + \bar{u}_{kj})\, z(i', j'') & & \\ & \sum_{\substack{j''\in V'':\\ \{i',j''\}\in E'\cup E''_h}} z(i', j'') = 1 & & i' \in V' \\ & \sum_{\substack{i'\in V':\\ \{i',j''\}\in E'\cup E''_h}} z(i', j'') = 1 & & j'' \in V'' \\ & z(i', j'') \geq 0 & & \{i', j''\} \in E' \cup E''_h \end{aligned}$$

whose dual is

$$\begin{aligned} \max \ & \sum_{i'\in V'} y(i') + \sum_{j''\in V''} y(j'') & & \\ & y(i') + y(j'') \leq \bar{u}_{ik} + \bar{u}_{kj} & & k \in K(i, j),\ \{i', j''\} \in E' \\ & y(i') + y(i'') \leq 0 & & \{i', i''\} \in E''_h. \end{aligned}$$

These constraints, for all $h \in V$, have to be inserted into (48) so that the compact extended model of (48) is

$$\begin{array}{ll} \min \ \displaystyle\sum_{e\in E} u_e & \\ \displaystyle\sum_{i'\in V'} y_h(i') + \sum_{j''\in V''} y_h(j'') \geq 1 & h \in V \\ y_h(i') + y_h(j'') \leq u_{ik} + u_{kj} & k \in K(i,j),\ \{i', j''\} \in E',\ h \in V \\ y_h(i') + y_h(i'') \leq 0 & \{i', i''\} \in E''_h,\ h \in V \\ u_e \geq 0 & e \in E. \end{array} \tag{49}$$

This formulation has $2\,n^2 + m$ variables and at most $n^2\,(n-1)^2/4 + n\,(n-1) + n + m$ constraints. The $O(n^4)$ comes from all possible alternating paths of two arcs. An exact count of all alternating paths leads to the formula $\sum_{i\in V} d_i^W\, d_i^B$ with d_i^W the degree of vertex i for the white edges and d_i^B for the black edges.

Again, we need the dual of (49), i.e., the compact equivalent of (46), to retrieve the cycles and possibly imposing integrality on the corresponding variables. The dual, obtained after modifying (49) by multiplying all variables $y_h(i')$ by -1, is again a special multi-commodity flow problem on the bipartite graph G' modified to have multiple parallel arcs $(i', j'')_k$ for each $k \in K(i,j)$. For each $h \in V$ there is a flow ξ^h_{ikj} on the arc $(i', j'')_k$ and a flow ξ^h_i on the arc (i', i''). The flow ξ^h_h is set to 0. All nodes in V' are sources of a flow ζ^h and all nodes in V'' are sinks of a flow ζ^h. The flow ξ^h_{ikj} originates from these sources. The arcs (i', i'') have unbounded capacity and there is no upper bound on ξ^h_i, There are no capacity bounds on the arcs $(i', j'')_k$ either but the flows ξ^h_{ikj} have a peculiar constraint. The arc $(i', j'')_k$ is associated to the two edges $\{i,k\}$ and $\{k,j\}$ in E and the flow ξ^h_{ikj} is 'counted' both for the edge $\{i,k\}$ and for $\{k,j\}$ and it is the edges $\{i,j\} \in E$ which have a unity capacity bound. Hence the capacity bounds are

$$\sum_{h\in V}\sum_{k\in V} \xi^h_{ijk} \leq 1, \qquad \{i,j\} \in E$$

$$\sum_{h\in V}\sum_{k\in V} \xi^h_{kij} \leq 1, \qquad \{i,j\} \in E$$

The objective function is the maximization of $\sum_h \zeta^h$.

14 Robust Knapsack

The Robust Knapsack Problem (RKP) is described as follows: there are given n items of value v_j, $j := 1, \ldots, n$ and an integer $m \leq n$. For each item j a nominal weight w_j is given. The weight of each item can be increased by a quantity $\overline{w}_j \geq 0$ to the weight $w_j + \overline{w}_j$. The knapsack capacity is K. A subset $J \subset [n]$ is feasible if, for any subset $S \subset J$ of cardinality at most m, the following inequality is satisfied

$$\sum_{j\in J} w_j + \sum_{j\in S} \overline{w}_j \leq K$$

This is equivalent to saying that a subset J is feasible if and only if the previous inequality is satisfied for $S = J$ if $|J| \leq m$ and, if $|J| > m$, for the subset $S \subset J$ consisting of the m items in J with the largest $\overline{w}_j$. The RKP consists in finding the feasible subset J with largest value.

The RKP was first introduced in [6] in a more restricted version. The given formulation is due to [40]. The RKP is clearly NP-hard. It is also weakly NP-hard as the normal Knapsack problem since a pseudopolynomial dynamic programming algorithm is available [41]. For a recent survey see also [41].

The following ILP model to solve the Robust Knapsack problem was proposed in [7]

$$\begin{aligned} \max \quad & \sum_{j\in[n]} v_j\, x_j \\ & \sum_{j\in[n]} w_j\, x_j + \sum_{j\in[n]} t_j + m\, r \le K \\ & t_j + r \ge \overline{w}_j\, x_j \qquad\qquad j \in [n] \\ & x_j \in \{0,1\},\ t_j \ge 0,\ r \ge 0 \qquad j \in [n]. \end{aligned} \tag{50}$$

Later a branch-and-cut model was developed by [23] who noted that robustness can be enforced to a normal knapsack model via the following *robustness cuts*

$$\sum_{j\in[n]} w_j\, x_j + \sum_{j\in S} \overline{w}_j\, x_j \le K, \qquad \forall S \subseteq N : |S| \le m. \tag{51}$$

We will now show how the model (50) is nothing but the compact extended formulation of the large-scale LP using constraints (51). Given the current solution x^*, the separation of robustness cuts is straightforward and it can be solved also by the following LP problem

$$\begin{aligned} \max \quad & \sum_{j\in[n]} (\overline{w}_j\, x_j^*)\, z_j \\ & \sum_{j\in[n]} z_j \le m \\ & z_j \le 1 \qquad j \in [n] \\ & z_j \ge 0 \qquad j \in [n] \end{aligned} \tag{52}$$

and asking whether the optimal value is not larger than $K - \sum_{j\in[n]} w_j\, x_j^*$. The dual of (52) is the following LP

$$\begin{aligned} \min \quad & m\, r + \sum_{j\in[n]} t_j \\ & r + t_j \ge \overline{w}_j\, x_j^* \qquad j \in [n] \\ & r \ge 0, t_j \ge 0 \qquad j \in [n] \end{aligned} \tag{53}$$

and the feasibility condition for x^* requires $m\, r + \sum_{j\in[n]} t_j \le K - \sum_{j\in[n]} w_j\, x_j^*$. Hence we may replace (51) with these constraints and get, after 'freeing' the values x_j^*,

$$\begin{aligned} \max \quad & \sum_{j\in[n]} v_j\, x_j \\ & m\, r + \sum_{j\in[n]} t_j \le K - \sum_{j\in[n]} w_j\, x_j \\ & r + t_j \ge \overline{w}_j\, x_j \qquad\qquad j \in [n] \\ & r \ge 0, t_j \ge 0, x_j \ge 0 \qquad j \in [n] \end{aligned} \tag{54}$$

which is exactly the model (50).

15 Job-Shop

In the job-shop problem, a set M of m machines and a set J of n jobs are given. Every job $j \in J$ consists of a sequence of $n(j)$ operations $O_1^j \to \cdots \to O_k^j \to \cdots \to O_{n(j)}^j$ and each operation O_k^j has to be processed without preemption on the machine $\mu(j,k)$ with a known processing time $q(j,k) > 0$. Let L bet the set of all operations, and $\ell = |L| = \sum_{j \in J} n(j)$. A *feasible schedule* of the jobs in J is a set of completion times $t(j,k)$ associated to each operation O_k^j such that: (i) the job precedence relations of the operations are respected and (ii) operations associated to the same machine do not overlap in time. It is not excluded that a machine can process more than one operation for the same job. The time $t(j,n(j))$ is the completion time of the job j.

The job-shop problem considered in [36] has a total cost given by the sum of the costs for the single jobs. The cost of a job is given by a generic function

$$f_j : R \to R \cup \{+\infty\}, \qquad t \mapsto f_j(t) \tag{55}$$

which assigns a penalty to the completion time t. The function $f_j(t)$ takes on value $+\infty$ when its argument t is an infeasible completion time for job j (e.g., because of release dates, deadlines, ecc.). Hence the cost of a feasible schedule is defined as the following separable objective function

$$\sum_{j \in J} f_j(t(j,n(j))). \tag{56}$$

To solve this type of job-shop problem, a column-generation model is defined based on *scheduling patterns*. A scheduling pattern p defines, for each operation k of a job j, its starting time $s_p(j,k)$ and its ending time $t_p(j,k)$. Let $t(p) := t_p(j,n(j))$ be the completion time of the last operation of job j for the pattern p. The cost of a pattern is given by

$$c(p) = f_j(t(p)). \tag{57}$$

This model requires the definition of a value $\bar{T}$ for the time horizon. Let us denote by P^j the set of patterns for job j with $t(p) \le \bar{T}$. To each pattern p in P^j, an $(m\,\bar{T})$-dimensional binary vector a^p is associated, with m fields of length $\bar{T}$, one for each machine $h \in M$. The t-th entry of the h-th field $a^p_{h,t}$ is 1 if and only if the operation of the job which must be executed by machine h is being processed in the time slot $[t-1,t]$.

A binary variable x_p is associated to each pattern p of P^j, with the meaning that $x_p = 1$ if and only if the job j is scheduled according to the pattern p. Then the job-shop problem with total cost objective may be formulated as the following binary LP:

$$\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{p \in P^j} c(p)\, x_p & & \\
& \sum_{p \in P^j} x_p = 1 & & j \in J \\
& \sum_{j \in J} \sum_{p \in P^j} a^p_{h,t}\, x_p \le 1 & & h \in M, \quad t = 1, \ldots \bar{T} \\
& x_p \in \{0,1\} & & p \in P^j, \quad j \in J.
\end{aligned} \tag{58}$$

Let us denote by $u(j)$ and $v(h,t) \le 0$ the dual variables of ($\overline{58}$). The pricing problem can be solved for each job j by a forward dynamic-programming procedure (see [36] for details). Let $V(k,t)$ represent the minimum

reduced cost of a pattern consisting of the first k operations and completing the k-th operation *within* t. Initialize $V(0,t) := 0$ for each t and the other values as $V(k,t) := +\infty$. The values $V(k,t)$ can be recursively computed as

$$V(k,t) = \min\{V(k,t-1), V(k-1,t-q(k)) + f_{jk}(t) - \sum_{\tau=t-q(k)+1}^{t} \hat{v}(\mu(k),\tau)\} \tag{59}$$

where the two terms in the above expression represent the minimum reduced cost of patterns which complete the k-th operation before t and exactly at time t, respectively. As a consequence, for any time t, the minimum reduced cost of a pattern $p \in P^j$ with $t(p) \le t$ is given by $V(n(j),t) - \hat{u}(j)$.

The dynamic-programming recursion formula (59) allows rewriting the dual of ($\overline{58}$) in a pseudo-compact extended way as shown in detail in [36]. The size of this formulation is $(\ell + m)\,\bar{T} + n$ variables and $2\,\ell\,\bar{T} + n$ constraints. In turn its dual, i.e., the compact equivalent of ($\overline{58}$), is a particular flow problem, defined on the following network $G = (N, E)$. The node set N is given by

$$N = \left\{(j,k,t) : j \in J,\, 0 \le k \le n(j),\, 0 \le t \le \bar{T}\right\}.$$

The nodes can be partitioned into levels $L(j,k)$, $j \in J$ and $0 \le k \le n(j)$ where each level $L(j,k)$, $k > 0$, contains a node (j,k,t) for each possible completion time t of the k-th operation of j and each level $L(j,0)$ contains a node $(j,0,t)$ for each possible starting time of the first operation.

The graph contains arcs of two types denoted as E_0 and E_1 and defined as

$$E_0 := \left\{((j,k,t-1),(j,k,t)) : j \in J,\, 0 \le k \le n(j),\, 0 < t \le \bar{T}\right\}$$

$$E_1 := \left\{((j,k-1,t-q(j,k)),(j,k,t)) : j \in J,\, 0 < k \le n(j),\, 0 \le t \le \bar{T}\right\}.$$

Then $E = E_0 \cup E_1$. The network G has n connected components, one for each job. In the network there are n source-sink pairs. The sources are the nodes $s_j := (j,0,0,)$ and the sinks are the nodes $d_j := (j,n(j),\bar{T})$. The idea is to send one flow unit from each source s_j to each sink d_j. If the flow is integral it gives rise to a path $s_j \to d_j$ which can be interpreted as a scheduling of the job j.

Let $\xi^0(j,k,t)$ be the flow associated to the arc $((j,k,t-1),(j,k,t)) \in E_0$ and let $\xi^1(j,k,t)$ be the flow associated to the arc $((j,k-1,t-q(j,k)),(j,k,t)) \in E_1$.

Each arc $(j,k-1,t-q(j,k)),(j,k,t)) \in E_1$ is also associated to the machine $\mu(k)$, so that we may partition E_1 into subsets $E_1(m)$ for each machine m. Let $E_1(m,t)$ the set of arcs $(j,k-1,t'-q(j,k)),(j,k,t'))$ in $E_1(m)$ such that $t'-q(j,k) < t \le t'$. The machine constraints require the total flow on the arcs $E_1(m,t)$ to be bounded by 1. In conclusion, the compact equivalent of ($\overline{58}$) is

$$\begin{aligned}
\min \quad & \sum_{(j,k,t)} f_{jk}(t)\,\xi^1(j,k,t) \\
& \xi^j \in \Phi(s_j, d_j, 1) \qquad j \in J \\
& \sum_{e \in E^1(m,t)} \xi^j_e \le 1 \qquad m \in M, 0 \le t \le \bar{T}.
\end{aligned}$$

16 Protein fold comparison

A protein consists of a sequence of residues, also called amino acids, linearly arranged along its backbone. In the aqueous solution in which it naturally occurs, the protein quickly folds into a very compact form fully

determined by the sequence of its amino acids. The folding is often indicative of the protein's function and hence it is valuable to have an abstract description that captures the important features of this folding. One such description is the protein's *contact map*.

A contact map is a graph whose vertices are the residues of the protein. The vertex set is linearly ordered by the order in which the residues occur on the backbone. There is an edge between two vertices whenever the distance between the residues in the folded protein is smaller than a certain threshold (e.g., 5 Å). Any two such residues are said to be *in contact*.

In order to compute the similarity of two contact maps, one can use the *contact map overlap* (CMO) measure [27]. Given two proteins, their CMO is found by determining the best *alignment* (i.e., a particular type of matching) of the residues of one proteins with those of the other. The value of an alignment is given by the number of pairs of residues in contact in the first protein which are aligned with pairs of residues that are also in contact in the second protein. The optimal alignment has maximum value, so that the CMO is the largest number of common contacts.

The contact map overlap problem can be formally stated as follows: given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ in which the vertices are linearly ordered, find two isomorphic, edge-induced, subgraphs such that (i) the isomorphism map between the vertices of the two subgraphs is monotonic (i.e., if a vertex i is mapped to a vertex j, then no vertex following i can be mapped to vertex preceding j) and (ii) the number of edges in each subgraph is maximum. A monotonic isomorphism between the vertices of two subgraphs is also called a *non-crossing* alignment.

An IP formulation of CMO is the following [35]. Let $V_1 = [n_1]$ and $V_2 = [n_2]$, and denote each edge e in E_1 or E_2 by an ordered pair (u, v), with $u < v$. Define binary variables x_{ij}, for $i \in V_1$ and $j \in V_2$, representing the alignment lines, i.e., the pairs of residues aligned in the two proteins. Furthermore, let z_{ef} be binary variables that are set to 1 whenever two contacts $e \in E_1$ and $f \in E_2$ are in common for an alignment. Let $\mathcal{M}$ denote the collection of all sets of mutually incompatible alignment lines (i.e., each element F of $\mathcal{M}$ is a set of lines, and, for any $l', l'' \in F$, with $l' \neq l''$, there is no feasible alignment containing both l' and l''). The IP formulation is then:

$$
\begin{aligned}
\max \quad & \sum_{\substack{(i,j)\in E_1 \\ (u,v)\in E_2}} z_{(i,j)(u,v)} & & \\
& \sum_{(i,j)\in E_1} z_{(i,j)(u,v)} \leq x_{iu}, \qquad \sum_{(j,i)\in E_1} z_{(j,i)(u,v)} \leq x_{iv} & & i \in V_1,\ (u,v) \in E_2 \\
& \sum_{(u,v)\in E_2} z_{(i,j)(u,v)} \leq x_{iu}, \qquad \sum_{(v,u)\in E_2} z_{(i,j)(v,u)} \leq x_{iu} & & u \in V_2,\ (i,j) \in E_1 \\
& \sum_{[i,j]\in F} x_{ij} \leq 1 & & F \in \mathcal{M} \\
& x, z \in \{0,1\}. & &
\end{aligned}
\tag{60}
$$

The constraints $\sum_{[i,j]\in F} x_{ij} \leq 1$, $F \in \mathcal{M}$, are called *clique inequalities*. The name comes from considering a *conflict graph* for the alignment lines. In this graph, every alignment line $[i, j]$ is a vertex, and two alignment lines are connected if they are not compatible. A clique in in the conflict graph is then a set of mutually incompatible alignment lines.

The LP $(\overline{60})$ can be solved in polynomial time thanks to a separation algorithm for the clique inequalities which calls for the longest path in a directed grid of size $n_1 \times n_2$ [38]. The nodes of the grid correspond to alignment lines of the CMO. Each node (i, j) of the grid is given a weight, equal to x_{ij}. The length of a

path in the grid is the sum of the weights of the path nodes. The most violated clique inequality is found by taking the longest path in R from $(1, n_2)$ to $(n_1, 1)$ and checking if its length is greater than 1. As we have seen this kind of constraints can be easily embedded into (60). Let y_{ij} be the length of the longest path from $(1, n_2)$ to (i, j). Then the compact extended reformulation of $(\overline{60})$ is obtained by replacing the clique inequalities with the constraints (see [12, 13])

$$\begin{aligned} y_{n_1,1} &\leq 1 \\ y_{1,n_2} &= x_{1,n_2} \\ y_{i,j} - y_{(i-1),j} &\geq x_{ij} \qquad i \in V_1 \setminus \{1\},\ j \in V_2 \\ y_{i,j} - y_{i,(j+1)} &\geq x_{ij} \qquad i \in V_1,\ j \in V_2 \setminus \{n_2\} \end{aligned} \tag{61}$$

The size of this formulation is $2\,n_1\,n_2 + m_1\,m_2$ variables and $2\,(n_1\,m_2 + n_2\,m_1 + n_1\,n_2) - (n_1 + n_2) + 2$, where $m_i = |E_i|$ for $i = 1, 2$.

17 Conclusions

Compact formulations can often provide a valid alternative to exponential formulations that require separation and/or pricing procedures. We have surveyed several examples of problems for which compact formulations have been proposed in the literature. Furthermore, we have described new compact models for other problems.

References

1. Appelgren L.,"A column generation algorithm for a ship scheduling problem", *Transportation Science*, **3**, 53–68 (1969).
2. Barahona F.,"On cuts and matchings in planar graphs", *Mathematical Programming*, **60**, 53–68 (1993).
3. Barahona F., M. Jünger, and G. Reinelt,"Experiments in quadratic 0-1 programming", *Mathematical Programming*, **44**, 127–137 (1989).
4. Barnhart C., E. L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh and P. H. Vance,"Branch-and-Price: Column Generation for Solving Huge Integer Programs", *Operations Research*, **46:3**, 316–329 (1998).
5. Berman H.M., J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne,"The protein data bank", *Nucleic Acids Research*, **28**, 235–242 (2000).
6. Bertsimas D., and M. Sim,"Robust discrete optimization and network flows", *Mathematical Programming*, **98**, 49–71 (2003).
7. Bertsimas D., and M. Sim,"The price of robustness", *Operations Research*, **52**, 35–53 (2004).
8. Caprara A.,"Sorting permutations by reversals and Eulerian cycle decompositions", *SIAM Journal on Discrete Mathematics*, **12**, 91–110 (1999).
9. Caprara A., R.D. Carr, G. Lancia, B. Walenz and S. Istrail,"1001 optimal PDB structure alignments: integer programming methods for finding the maximum contact map overlap", *Journal of Computatinal Biology*, **11**, 27–52 (2004).
10. Caprara A., G. Lancia and S.K. Ng,"Sorting permutations by reversal through branch-and-price", *Informs Journal on Computing*, **13**, 224–244 (2001).
11. Caprara A., A. Panconesi and R. Rizzi,"Packing cycles in undirected graphs", *Journal of Algorithms*, **48**, 239–256 (2003).

12. Carr R.D., and G. Lancia,"Compact vs exponential-size LP relaxations", *Operations Research Letters*, **30**, 57–65 (2002).
13. Carr R.D., and G. Lancia,"Compact optimization can outperform separation: A case study in structural proteomics", *4OR*, **2**, 221–233 (2004).
14. de Carvalho J.M.V.,"Exact solutions of bin-packing problems using column generation and branch-and-bound", *Annals of Operations Research*, **86**, 629–665 (1999).
15. de Carvalho J.M.V.,"LP models for bin packing and cutting stock problems", *European Journal of Operational Research*, **141**, 253–273 (2002).
16. Chvátal V.,"On certain polytopes associated with graphs", *Journal of Combinatorial Theory Ser. B*, **18**, 138–154 (1975).
17. Conforti M., G. Cornuéjols and G. Zambelli,"Extended formulations in combinatorial optimization", *4OR*, **8**, 1–48 (2010).
18. Cook W.J., W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, John Wiley & Sons, New York (1998).
19. Dantzig G. B., R. Fulkerson and S. M. Johnson,"Solution of a large-scale traveling salesman problem", *Operations Research*, **2**, 393–410 (1954).
20. De Simone C., and G. Rinaldi,"A cutting plane algorithm for the max-cut problem", *Optimization Methods and Software*, **3**, 195–214 (1994).
21. Fiorini S., S. Massar, S. Pokutta, H. Raj Tiwary and R. de Wolf, "Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds", in: *44th ACM Symposium on Theory of Computing (STOC 2012)*, New-York (NY), USA, 19-22 May 2012 (2012).
22. Fischetti M., G. Lancia, and P. Serafini,"Exact algorithms for minimum routing cost trees", *Networks*, **39**, 1-13 (2002).
23. Fischetti M., and M. Monaci,"Cutting plane versus compact formulations for uncertain (integer) linear programs", *Mathematical Programming Computation*, **4**, 239–273 (2012).
24. Gerards A.M.H., and A. Schrijver,"Matrices with the Edmonds-Johnson property", *Combinatorica*, **6**, 365–379 (1986).
25. Gilmore P.C., and R.E. Gomory,"A linear programming approach to the cutting stock problem", *Operations Research*, **9**, 849-859 (1961).
26. Gilmore P.C., and R.E. Gomory,"A linear programming approach to the cutting stock problem - II", *Operations Research*, **11**, 863-888 (1963).
27. Goldman D., S. Istrail, and C. Papadimitriou, "Algorithmic aspects of protein structure similarity", in: *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, 512–522 (1999).
28. Grötschel M., and O. Holland,"Solution of large-scale travelling salesman problems", *Mathematical Programming*, **51(2)**, 141–202 (1991).
29. Grötschel M., M. Jünger, and G. Reinelt, "Calculating exact ground states of spin glasses: A polyhedral approach", in: *Heidelberg Colloquium on Glassy Dynamics*, 325–353, Springer, Berlin (1987).
30. Grötschel M., L. Lovász, and A. Schrijver,"The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica*, **1**, 169-197 (1981).
31. Grötschel M., L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer, Berlin, Germany (1988).
32. Hu T. C.,"Optimum communication spanning trees", *SIAM J. Comp.*, **3**, 188–195 (1974).
33. Kaibel V., "Extended formulations in combinatorial optimization", arXiv preprint arXiv:1104.1023 (2011).
34. Kaibel V. and K. Pashkovich, "Constructing extended formulations from reflection relations", in: *Integer Programming and Combinatorial Optimization XV, Lecture Notes in Computer Science 6655*, O. Günlük, G. Woeginger eds, Springer, 287–300 (2011).

35. Lancia G., R.D. Carr, B. Walenz, and S. Istrail, "101 optimal PDB structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem", in: *Proceedings of 5th ACM Intl. Conf. on Computational Molecular Biology (RECOMB)*, 193–202 (2001).
36. Lancia G., F. Rinaldi, and P. Serafini,"A time-indexed LP-based approach for min-sum job-shop problems", *Annals of Operations Research*, **86**, 175–198 (2011).
37. Lancia G., and P. Serafini,"An effective compact formulation of the max cut problem on sparse graphs", *Electronic Notes in Discrete Mathematics*, **37**, 111–116 (2011).
38. Lenhof H.P., K. Reinert, and M. Vingron,"A polyhedral approach to RNA sequence structure alignment", *Journal of Computational Biology*, **5**, 517–530 (1998).
39. Martin K.,"Using separation algorithms to generate mixed integer model reformulations", *Operations Research Letters*, **10**, 119–128 (1991).
40. Monaci M., and U. Pferschy, "On the robust knapsack problem", in: *CTW*, 207–210 (2011).
41. Monaci M., U. Pferschy and P. Serafini,"Exact solution of the robust knapsack problem", *Computers & Operations Research*, **40**, 2625–2631 (2013).
42. Newman A., "Max cut", in: *Encyclopedia of Algorithms*, Kao, Ming-Yang ed, 1–99, Springer US (2008).
43. Padberg M., and G. Rinaldi,"A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems", *SIAM Review*, **33(1)**, 60–100 (1991).
44. Stoer M., and F. Wagner, "A simple mincut algorithm", in: *Proceedings of ESA 94, Lecture Notes in Computer Science 855*, 141-147, Springer, Berlin (1994).
45. Wu B.Y., G. Lancia, V. Bafna, K.M. Chao, R. Ravi, and C.Y. Tang,"A polynomial–time approximation scheme for minimum routing cost spanning trees", *SIAM J. Comp.*, **29**, 761–778 (1999).
46. Yannakakis M.,"Expressing combinatorial optimization problems by linear programs", *Journal of Computer and System Sciences*, **43**, 441–466 (1991).