MATHEURISTICS FOR COMBINATORIAL OPTIMIZATION PROBLEMS

Module 1 - Lesson 2

Prof. Maurizio Bruglieri
Politecnico di Milano

Although this may seem a paradox, all exact science is dominated by the idea of approximation.

Bertrand Russell (1872–1970)

Heuristics with approximation guarantee

- We want a guarantee that the heuristic algorithm provides a solution with value not much worse than the optimal one

Given an optimization problem P, where z_{opt} is the optimal value and z_A the value provided by the huristic, we call

- absolute error $E_A = |z_{opt} - z_A|$

- relative error $R_A = |z_{opt} - z_A| / |z_{opt}|$

Note $z_{opt} \neq 0$

Let *I* an instance (i.e. a particular case) for problem P

An algorithm A is *absolutely* approximated for problem P if and only if for each instance *I*

$$\left| z_{opt}(I) - z_A(I) \right| \le k$$

for a certain constant $k > 0$

Algorithm A is $g(n)$- approximated algorithm for problem P if and only if for each instance *I* of size *n*

$$\left| z_{opt}(I) - z_A(I) \right| \le g(n) \left| z_{opt}(I) \right|$$

Algorithm A is an ε- approximated algorithm for problem P if and only if for each instance *I*

$$\left| z_{opt}(I) - z_A(I) \right| \le \varepsilon \left| z_{opt}(I) \right|$$

for a certain constant $\varepsilon > 0$

An algorithm A is an *approximation scheme* for problem P ⇔ for each $\varepsilon > 0$ and for each instance I

$z_A(I) \leq (1+\varepsilon) z_{opt}(I)$ if P is a minimization problem

$z_A(I) \geq (1-\varepsilon) z_{opt}(I)$ if P is a maximization problem

An algorithm A is an *polynomial time approximation scheme* (PTAS) for problem P ⇔ it is an AS s.t. for each $\varepsilon > 0$ and for each instance I its computational complexity is bounded by a polynomial in $| I |$

An algorithm A is a *fully polynomial time approximation scheme* (FPTAS) for problem P ⇔ it is an AS s.t. for each instance I its computational complexity is bounded by a polynomial in $| I |$ and $1/\varepsilon$

Non approximable COPs

- Some NP-hard COPs are so hard that they cannot be even approximated

The TSP cannot be approximated for any $\varepsilon > 0$

Proof: Reduction from Hamiltonian Cycle Problem (HCP)
Consider any HCP instance i.e., a directed graph G=(N,A) with $|N| = n$ and consider the complete graph G' on N with the following costs:

$$c_{ij} = \begin{cases} 1 \ \ if \ (i,j) \ \in \ A \\ n\varepsilon \ \ \text{otherwise} \end{cases}$$

If G is Hamiltonian, the optimal solution of the TSP in G' would be n, otherwise $> n\varepsilon$
In the first case the approximated algorithm would provide a solution of value $\leq n\varepsilon$ while in the second case a solution of value $n\varepsilon$

How can one find the the limits of approximation for the approximated algorithms?

As we will se in the next examples three ingredients are necessary (consider e.g. minimum problems) :

- An upper bound, z_A , of the optimal solution value, obtained through the heuristic

- A lower bound z_{LB} of the optimal solution value, obtained for instance with a relaxation

- A function $f(z_{LB})$ non decreasing whose value is not lower than z_A in such a way that one obtains

$$z_{LB} \leq z^* \leq z_A \leq f(z_{LB}) \leq f(z^*)$$

Example 1

Integer knapsack

Consider the problem

$$z^* = \max \{c^T x : a\, x \le b,\ x \in Z^n_+\}$$

where $b, a_1, \ldots, a_n \in Z_+$ and for *hypothesis* $a_j \le b$, with $j = 1, \ldots, n$ and the relation $c_1 / a_1 \ge c_j / a_j$ holds for $j = 2, \ldots, n$

Greedy algorithm

1. Fill the knapsack with as many copies as possible of the object with the best ratio “cost over volume”

Consider the *greedy* solution $x^H = (\left\lfloor \frac{b}{a_1} \right\rfloor, 0, \ldots, 0)$

with value $z^H = c_1 \left\lfloor \frac{b}{a_1} \right\rfloor \leq z^*$

The solution of the linear relaxation provides an upper bound $z^{LP} = c_1\, b / a_1 \geq z^*$

From $a_1 \leq b$ it follows $\left\lfloor \frac{b}{a_1} \right\rfloor \geq 1$. Setting $\frac{b}{a_1} = \left\lfloor \frac{b}{a_1} \right\rfloor + f$, with $0 \leq f < 1$

One obtains $\left\lfloor \frac{b}{a_1} \right\rfloor \Big/ \frac{b}{a_1} = \frac{\left\lfloor \frac{b}{a_1} \right\rfloor}{\left\lfloor \frac{b}{a_1} \right\rfloor + f} \geq \frac{\left\lfloor \frac{b}{a_1} \right\rfloor}{\left\lfloor \frac{b}{a_1} \right\rfloor + \left\lfloor \frac{b}{a_1} \right\rfloor} = \frac{1}{2}$

Hence $z^H / z^* \geq z^H / z^{LP} = \frac{c_1 \left\lfloor \frac{b}{a_1} \right\rfloor}{c_1 \frac{b}{a_1}} = \frac{\left\lfloor \frac{b}{a_1} \right\rfloor}{\frac{b}{a_1}} \geq \frac{1}{2}.$

The algorithm is 1/2-approximated

Numerical example

$c = (20, 27, 9, 24, 6)$

$a = (5, \; 7, \; 4, \; 9, \; 3) \quad b=22$

$x^{\mathrm{H}} = (\lfloor 22/5 \rfloor,0,0,0,0) = (4,0,0,0,0) \quad z^{\mathrm{H}} = 80 \quad z^{LP} = 88$

$z^H / z^{LP} = 80/88 \approx 0{,}909$

$x^* = (0,3,0,0,0) \quad z^* = 81$

$z^H / z^* = 80/81 \approx 0.99$

$z^H / z^* \geq z^H / z^{LP} \geq 0.5$

In this case the error is about the 1.2 %. For no instance it will be greater than 100%.

Example 2

Set covering

Given $M=\{1,2,\ldots,m\}$ and a family n of subsets $S_j \subseteq M$, with $j \in N =\{1,2\ldots,n\}$. With each subset S_j is associated a cost c_j. We look for the set of susbsets with minimum cost whose union cover all elements of M.

Greedy algorithm

1. Order in L the subsets for non decreasing values of ratio "cost divided by the non covered elements that they can cover"

2. Repeat until all elements of M are covered

    remove from L next subset, S, in the given order;

    label as covered all non covered elements in S;

    update the order of L;

Numerical example

$M = \{1,2,\ldots,5\}$, $N = \{1,2,\ldots,6\}$ $\underline{c}^T = (4,6,10,14,5,6)$

$S_1 = \{3,5\}$ $S_2 = \{1,3,5\}$ $S_3 = \{1,2,5\}$ $S_4 = \{1,2,4\}$ $S_5 = \{1,4,5\}$ $S_6 = \{3,4\}$

Step 1: Ratio = (4/2, 6/3, 10/3,14/3, 5/3,6/2); Choose $S_{5;}$

Step 2: Ratio = (4/1, 6/1, 10/1,14/1, -- ,6/1); Choose $S_{1;}$

Step 3: Ratio = (-- , ∞ , 10/1,14/1, -- , ∞); Choose $S_{3;}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Incidence matrix

$S_5 \cup S_1 \cup S_3 = M$ $z_A = 19$

$S_3 \cup S_6 = M$ $z^* = 16$

Let $k = \max_j \{ |S_j| \}$, it is possible to prove:

the algorithm is $log(k)$-approximated

Example 3

Job assignment problem

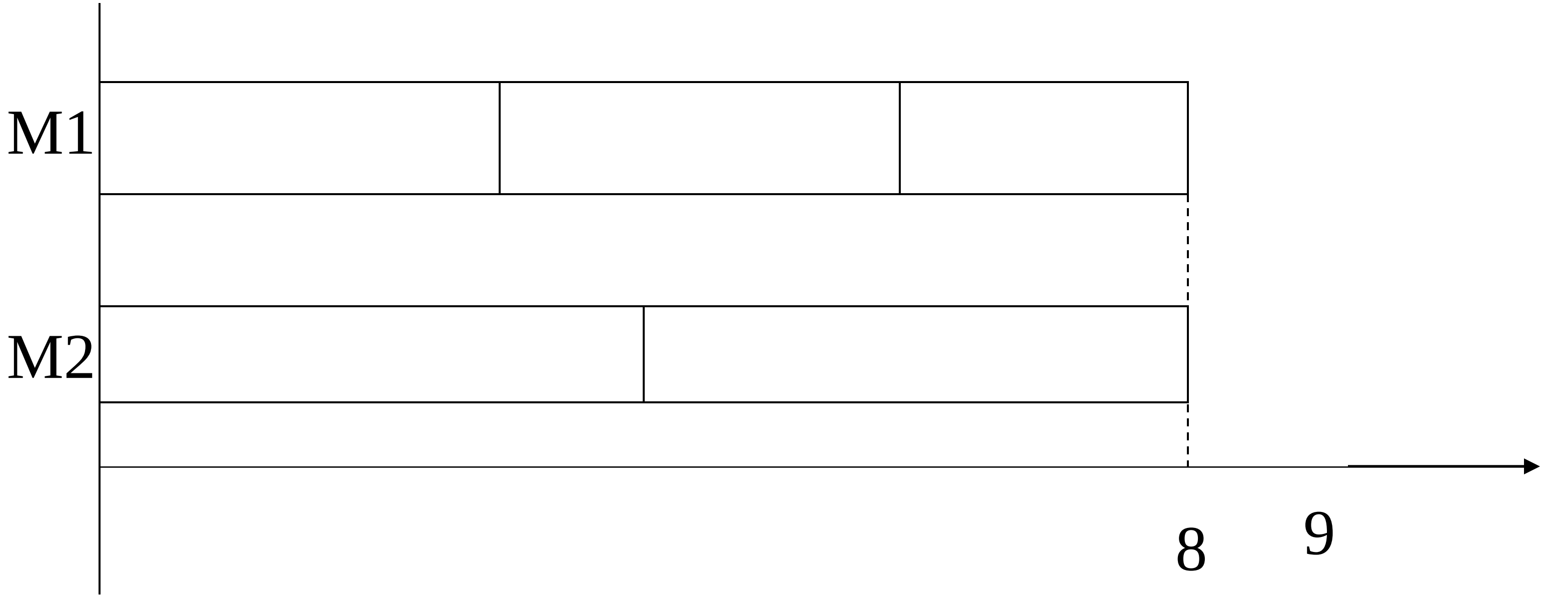
There are m identical machines and n jobs. Each job j, with $j=1,\ldots,n$, has to be processed from one the m machines for a processing time p_j. Every machine processes one job at the time. We want to minimize the completion time z^* of all jobs.

G*reedy* algorithm

1. Order the jobs in any order

2. Assign the jobs in the given order to the less loaded machine

Numerical example



Let z_A the *greedy* algorithm solution value.

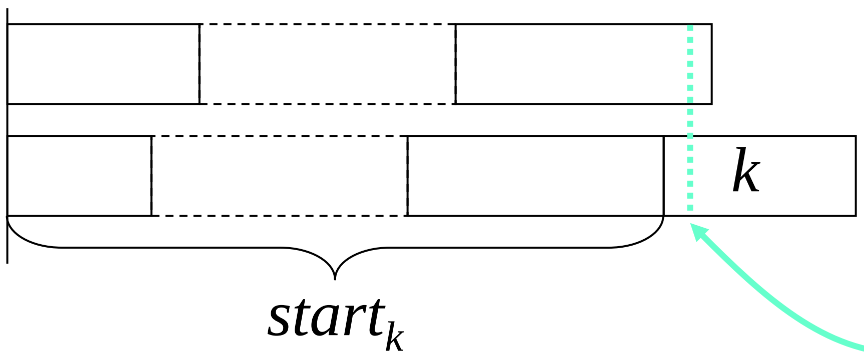
Evaluate the relative error for $m=2$

The quantities $LB = \frac{1}{2}\sum_{j=1}^{n} p_j$ and $\max_j \{p_j\}$ are lower bounds for z^*

Hence the relationship $LB \leq z^* \leq z_A$

Let k the index of last job executed and let $start_k$ its starting time. Hence $z_A = start_k + p_k$

Since the jobs are assigned to the less loaded machine, when k is assigned the other machine was working at least until the instant $start_k$



Hence

$$start_k \leq \frac{1}{2}\sum_{\substack{j=1 \\ j \neq k}}^{n} p_j = LB - p_k/2$$

$$z_A = start_k + p_k \leq LB + p_k/2 \leq z^* + z^*/2 = 3/2\, z^*$$

The algorithm is 1/2 -approximated

Example 4

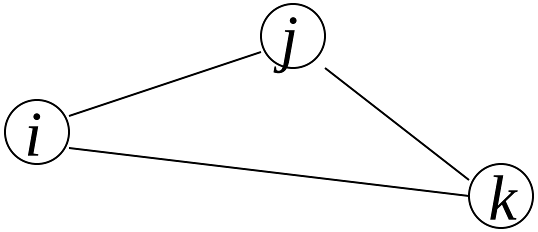
The symmetric TSP

Given a complete directed graph $G=(N,E)$ with non negative cost c_e for each edge $e=(i,j)$ of E, determine the minimum cost Hamiltonian cycle

The problem is NP-hard, but if in G the *triangular inequality* holds it is possible provide ε-approximated algorithms

<u>Triangular inequality</u>:

For every triplet of nodes i,j,k in N the following holds:



$$c_{ij} + c_{jk} \geq c_{ik}$$

Double tree algorithm

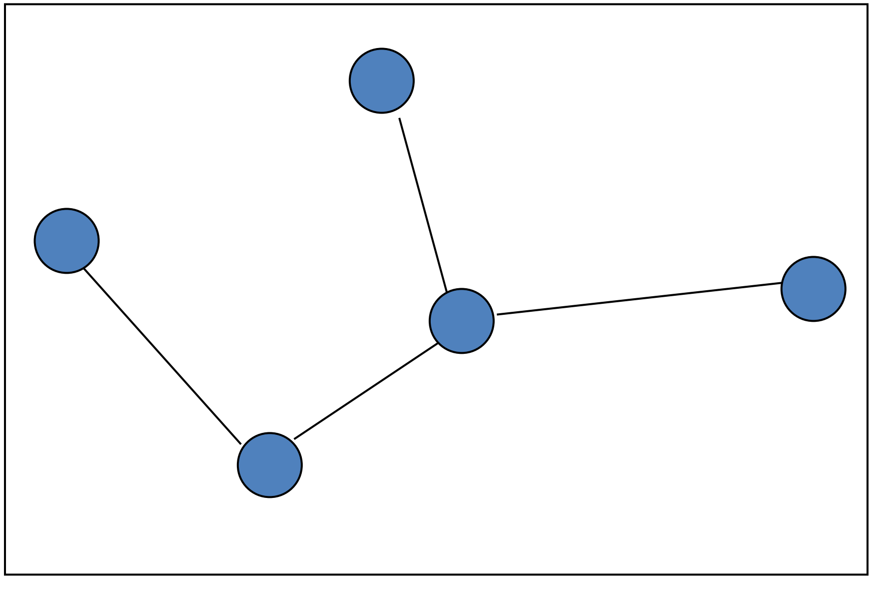
1. Build T^*, minimum cost spanning tree in G

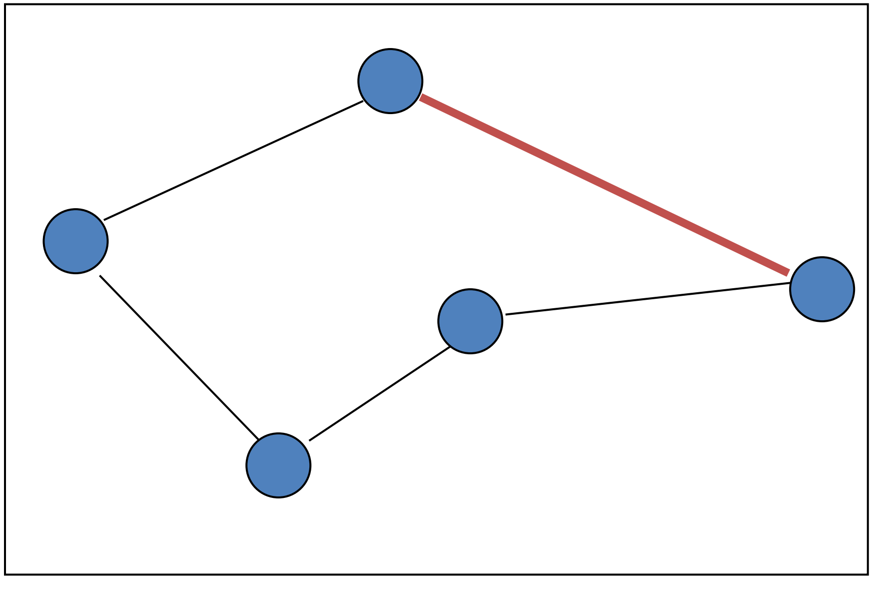
Observations:

Every Hamiltonian path is a spanning tree with cost $\geq c(T^*)$

Every Hamiltonian *cycle* is a Hamiltonian *path* with an additional edge

Therefore $c(T^*) \leq c(H^*)$



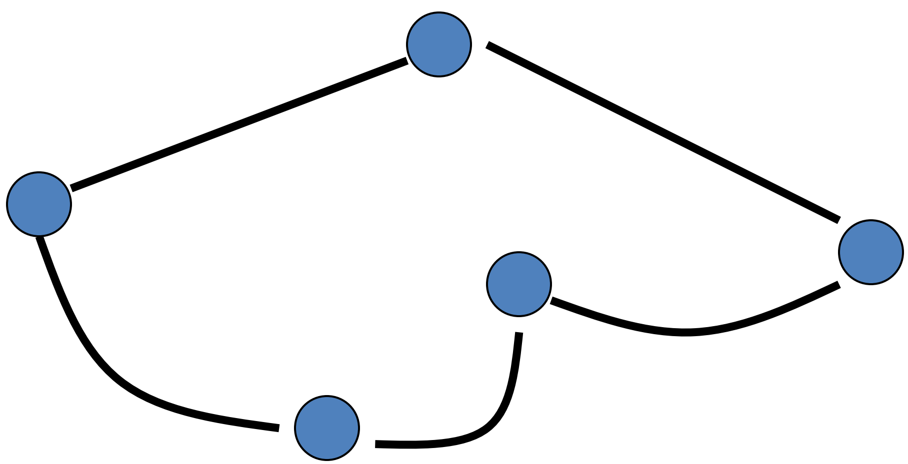


2. From the spanning tree we generate C visiting all nodes, also more than once.

Observations:

The cycle C can be transformed into an Hamiltonian cycle CH with cost not larger (thanks to the triangular inequality): $c(CH) \leq 2\, c(T^{*})$

Hence: $c(CH) \leq c(C) = 2\, c(T^{*}) \leq 2\, c(H^{*})$



The algorithm is

1-approximated

Approximated Matheuristics

Two main techniques for building an approximated algorithm from the Mathematical Programming formulation of a COP

Rounding its linear relaxation optimal solution

Primal-dual method

A rounding matheuristic for the MWNC

- Given an undirected graph G=(V,E) with a node cost function *c*, the Minimum Weight Node Cover Problem (MWNC) consists in finding a subset of vertices that covers i.e. touches each edge at least once and whose total cost is minimal.

$$\min z = \sum_{i=1}^{n} c_i x_i$$

$$x_i + x_j \geq 1 \; \forall [i,j] \in E$$

$$x_i \in \{0,1\} \text{ for } i = 1,..,n$$

- Let $\tilde{x}$ the optimal solution of the linear relaxation:
  $\forall \, [i,j] \in E, \text{either } \widetilde{x_i} \geq 0.5 \text{ or } \widetilde{x_j} \geq 0.5$

- Therefore if we round up every $\widetilde{x_i} \geq 0.5$ and to 0 the others we obtain a feasible solution

- The value of this feasible solution, $\hat{z}$ is $\leq 2\tilde{z}$ being $\tilde{z}$ the optimal value of the LR

- Hence, $\hat{z} \leq 2\tilde{z} \leq 2z^*$, i.e., this is a 2-approximated algorithm!

Rounding approximated matheuristic

- **General schema** (min problem):
  1. Solve the linear relaxation of the COP formulation → relaxed solution $\tilde{x}$
  2. From $\tilde{x}$ build the integer feasible solution $\hat{x}$ ensuring of not worsening too much the objective function → $z(\hat{x}) \leq (1 + \varepsilon) z(\tilde{x})$
  3. Thus $z(\hat{x}) \leq (1 + \varepsilon) z(\tilde{x}) \leq (1 + \varepsilon) z^*$

Rounding approximated matheuristic for set covering

- Linear relaxation of SC formulation:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
\text{subject to} \quad & \sum_{S:\ e \in S} x_S \geq 1, \qquad e \in U \\
& x_S \geq 0, \qquad\qquad S \in \mathcal{S}
\end{aligned}
$$

- Let *f* = frequency of the most frequent element in the sets

Rounding approximated matheuristic for set covering

Algorithm1 (Set Covering via LP-rounding):

1. Find an optimal solution to the LP-relaxation.
2. Pick all sets *S* for which $x_S \geq 1/f$ in this solution.

Theorem: Algorithm 1 achieves a *f* approximation factor for the set covering
Proof: Let *C* be the collection of picked sets. Consider an arbitrary element *e*.
Since *e* is in at most *f* sets ⇒ $\exists\ S$ with $e \in S$: $\tilde{x}_S \geq 1/f$
Thus, *e* is covered by *C*, and hence *C* is feasible set cover.
Since the rounding increases $\tilde{x}_S$, for each set *S* ∈ *C*, by a factor of at most *f*.
⇒ the cost of *C* is at most *f* times the cost of the fractional cover

- **Remark:** Algorithm 1 generalizes the rounding algorithm of the MWNC since the latter is a SC where each element (edge) can be only in two sets (each one corresponding to its ending vertices, since each set is associated with a node, and its elements are its incident edges.)

Primal-dual based approximated matheuristics

- Primal:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \qquad i = 1, \ldots, m \\
& x_j \geq 0, \qquad\qquad\quad\;\; j = 1, \ldots, n
\end{aligned}
$$

- Dual:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{m} b_i y_i \\
\text{subject to} \quad & \sum_{i=1}^{m} a_{ij} y_i \leq c_j, \qquad j = 1, \ldots, n \\
& y_i \geq 0, \qquad\qquad\quad\;\; i = 1, \ldots, m
\end{aligned}
$$

Primal-dual based approx. matheuristics

- **Theorem (Complementary Slackness Conditions)**
  Let *x* and *y* be primal and dual feasible solutions, respectively.
  Then, *x* and *y* are both optimal iff all of the following conditions are satisfied:

  Primal complementary slackness conditions
  For each $1 \le j \le n$*: either* $x_j = 0$ *or* $\sum_{i=1}^{m} a_{ij} y_i = c_j$*; and*
  Dual complementary slackness conditions
  For each $1 \le i \le m$*: either* $y_i = 0$ *or* $\sum_{j=1}^{n} a_{ij} x_j = b_i$*.*

- Most of the primal–dual approximated algorithms run by ensuring only one set of conditions and suitably relaxing the other (for α = 1 the primal conditions are imposed, while for β = 1 the dual ones):

  Primal complementary slackness conditions
  Let $\alpha \ge 1$.
  For each $1 \le j \le n$: either $x_j = 0$ or $c_j/\alpha \le \sum_{i=1}^{m} a_{ij} y_i \le c_j$.

  Dual complementary slackness conditions
  Let $\beta \ge 1$.
  For each $1 \le i \le m$: either $y_i = 0$ or $b_i \le \sum_{j=1}^{n} a_{ij} x_j \le \beta \cdot b_i$,

Primal-dual based approx. matheuristics

- **Proposition 1**
  If *x* and *y* are primal and dual feasible solutions satisfying the conditions stated above then

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \beta \cdot \sum_{i=1}^{m} b_i y_i.$$

Proof:

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \alpha \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i$$

$$\leq \alpha\beta \sum_{i=1}^{m} b_i y_i \ .$$

Primal-dual based approx. matheuristics

- **General schema** (min problem):
  1. Start with a primal infeasible solution and a dual feasible solution (e.g., x = 0 and y = 0);
  2. Iteratively improve the feasibility of the primal solution, and the optimality of the dual solution, ensuring that in the end a primal feasible solution is obtained and all conditions stated above, with a suitable choice of α and β, are satisfied;
  3. The primal solution is always extended integrally, thus ensuring that the final solution is integral;
  4. The improvements to the primal and the dual go hand-in-hand: the current primal solution is used to determine the improvement to the dual, and vice versa;
  5. Finally, the cost of the dual solution is used as a lower bound on OPT, and by Proposition 1, the approximation guarantee of the algorithm is αβ.

Primal-dual based approx. matheuristic for the set covering

- We obtain a *f*-approximated algorithm setting α=1 and β=*f*

- **Primal**:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
\text{subject to} \quad & \sum_{S:\ e \in S} x_S \geq 1, \qquad e \in U \\
& x_S \geq 0, \qquad S \in \mathcal{S}
\end{aligned}
$$

- **Dual**:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in U} y_e \\
\text{subject to} \quad & \sum_{e:\ e \in S} y_e \leq c(S), \qquad S \in \mathcal{S} \\
& y_e \geq 0, \qquad e \in U
\end{aligned}
$$

Primal-dual based approx. matheuristic for the set covering

- The complementary slackness conditions are:

Primal conditions

$$\forall S \in \mathcal{S} : x_S \neq 0 \Rightarrow \sum_{e:\ e \in S} y_e = c(S).$$

Set S will be said to be *tight* if $\sum_{e:\ e \in S} y_e = c(S)$. Since we will increment the primal variables integrally, we can state the conditions as: *Pick only tight sets in the cover.*

Dual conditions

$$\forall e : y_e \neq 0 \Rightarrow \sum_{S:\ e \in S} x_S \leq f$$

Since we will find a 0/1 solution for $\boldsymbol{x}$, these conditions are equivalent to: *Each element having a nonzero dual value can be covered at most f times.* Since each element is in at most f sets, this condition is trivially satisfied for all elements.

Primal-dual based approx. matheuristic for the set covering

Algorithm 2 (Set covering via Primal-Dual):

1. **Initialization:** $\boldsymbol{x} \leftarrow \mathbf{0}$; $\boldsymbol{y} \leftarrow \mathbf{0}$
2. Until all elements are covered, do:

   Pick an uncovered element, say e, and raise y_e until some set goes tight.

   Pick all tight sets in the cover and update $\boldsymbol{x}$.

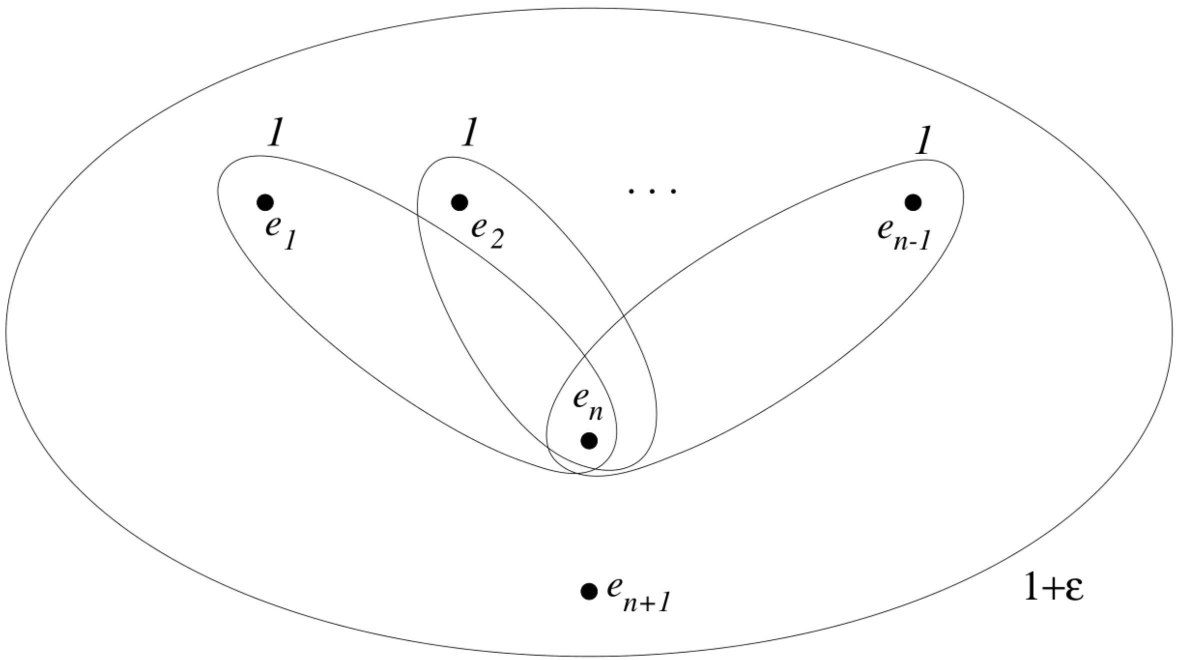
   Declare all the elements occurring in these sets as "covered".
3. Output the set cover $\boldsymbol{x}$.

Primal-dual based approx. matheuristic for the set covering

Theorem: Algorithm 2 achieves an approximation factor of *f*
Proof: Clearly there will be no uncovered elements and no overpacked sets at the end of the algorithm. Thus, the primal and dual solutions will both be feasible. Since they satisfy the relaxed complementary slackness conditions with $\alpha=1$ and $\beta=f$, by Proposition 1 the approximation factor is *f*

Remark: Although Algorithm 2 achieves the same approximation factor of Algorithm 1, it is generally faster since it does not require to solve any LP!

A tight example



Here, $\mathcal{S}$ consists of $n-1$ sets of cost 1, $\{e_1, e_n\}, \ldots, \{e_{n-1}, e_n\}$, and one set of cost $1+\varepsilon$, $\{e_1, \ldots, e_{n+1}\}$, for a small $\varepsilon > 0$. Since e_n appears in all n sets, this set system has $f = n$.

Suppose the algorithm raises y_{e_n} in the first iteration. When y_{e_n} is raised to 1, all sets $\{e_i, e_n\}$, $i = 1, \ldots, n-1$ go tight. They are all picked in the cover, thus covering the elements $e_1, \ldots, e_n$. In the second iteration, $y_{e_{n+1}}$ is raised to ε and the set $\{e_1, \ldots, e_{n+1}\}$ goes tight. The resulting set cover has a cost of $n + \varepsilon$, whereas the optimum cover has cost $1 + \varepsilon$. □

References

- M. O. Ball, (2011). Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16, pp. 21-38.
- T. Dokka, A.N. Letchford, M.H.Mansoor (2019), Anomalous Behaviour of Dual-Based Heuristics. *Optimization online*, pp.1-20.
- M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, CA, 1979.
- D.S. Hochbaum , (2002). Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research,* 140, pp. 291–321
- V.V. Vazirani. Approximation algorithms. Springer, 2003.
- J. Vygen, Approximation Algorithms for Facility Location Problems (Lecture Notes)