Approximation Algorithms

Integer linear programming: formulations, techniques and applications.

Santiago Valdés Ravelo
santiago.ravelo@inf.ufrgs.br

November, 2020
“...all exact science is dominated by the idea of approximation.”¹

Bertrand Russell

1 Introduction
2 Primal method
3 Dual method
4 Primal-dual method
5 Approximability classes
INTRODUCTION
Several combinatorial optimization problems are $NP$-hard.

Hence, there are no "efficient" algorithms to exactly solve them.

Although heuristics, metaheuristics and matheuristics can be used to solve these problems, generally those approaches have no guarantees on the solution feasibility and quality, nor the computational complexity required.
Approximation algorithms have the compromise of finding "efficiently" a "good quality" solution for any instance of a problem.

Generally, in this context "efficiently" means polynomial time.
Approximation algorithms. Definition

Given a minimization problem $\Pi$, an algorithm $A$ is an approximation algorithm if there exists $\alpha > 0$ such that, for any instance $I$ of $\Pi$:

$$\text{val}(A(I)) \leq \alpha \times \text{opt}_\Pi(I)$$

Where $\text{val}(A(I))$ is the value associated with the solution found by $A$ with instance $I$ and $\text{opt}_\Pi(I)$ is the value of an optimal solution of $\Pi$ with instance $I$.

For maximization problems the inequality is $\text{val}(A(I)) \geq \alpha \times \text{opt}_\Pi(I)$.
Job scheduling

The job scheduling receives an instance $I = \langle J, \tau, M \rangle$, where:

- $J$ is a finite set of jobs to be executed,
- $\tau : J \rightarrow \mathbb{Q}_+$ is a function over the jobs indicating the time required to execute each job,
- $M$ is a set of identical machines where the jobs will be executed.

The objective of the problem is to find and assignment $A : J \rightarrow M$, of the jobs to the machines that minimizes

$$\max_{m \in M} \left\{ \sum_{j \in J \land A(j) = m} \tau(j) \right\}.$$
Graham algorithm for job scheduling

Input: \( J, \tau : J \to \mathbb{Q}_+ \) and \( M \).

1. \( A \leftarrow \) null table indexed on elements of \( J \).

2. For each \( j \in J \), do:

3. \( \overline{m} \leftarrow m \in M \) that minimizes \( \sum_{j \in J \land A(j)=m} \tau(j) \).

4. \( A(\overline{j}) \leftarrow \overline{m} \).

5. Return \( A \).
Let $OPT$ be the value of an optimal solution, then:

$$OPT \geq \tau(j), \ \forall j \in J \quad \text{and} \quad OPT \geq \frac{1}{|M|} \times \sum_{j \in J} \tau(j)$$

(solution in which all machines have same time sum)

At the beginning of any iteration the time sum of the selected machine $\overline{m}$ satisfies:

$$\sum_{j \in J \land A(j)=\overline{m}} \tau(j) \leq \frac{1}{|M|} \times \sum_{m \in M} \sum_{j \in J \land A(j)=m} \tau(j) \leq 1 \times \sum_{j \in J} \tau(j) \leq OPT$$

At the end of the iteration, that value is increased in $\tau(\overline{j}) \leq OPT$.

Therefore, at the end of any iteration, the time sum of every machine is at most $2 \times OPT$. 

Theorem. Graham approximation algorithm is a polynomial time $2$-approximation algorithm for the job scheduling problem.

Note 1. Graham algorithm actually guarantees a $2 - \frac{1}{|M|}$-approximation ratio (prove as exercise).

Note 2. If the jobs are previously sorted in non-decreasing time, then Graham algorithm guarantees a $\frac{4}{3}$-approximation ratio (prove as exercise).
The metric TSP receives an instance $I = \langle G = (V, E), \ell \rangle$, where:

- $G = (V, E)$ is a complete undirected graph,
- $\ell : E \to \mathbb{Q}_+$ is a length function over the edges satisfying the triangular inequality: $\forall u, v, w \in V, \ell(uv) + \ell(uw) \geq \ell(vw)$.

The objective of the problem is to find a cycle $C \subseteq G$ that passes through every node of $G$ and minimizes $\sum_{e \in C} \ell(e)$. 
If $T$ is a minimum spanning tree of $G$, then: $\sum_{e \in T} \ell(e) \leq \text{OPT}$. 

If $M$ is a minimum perfect matching considering the nodes with odd degree in $T$, then:

$$\sum_{e \in M} \ell(e) \leq \frac{1}{2} \times \text{OPT}$$

All node degrees of $T' = T \cup M$ are even and there exists an Eulerian circuit $C' = u_1u_2...u_nu_1$ of $T'$, containing all nodes of $G$.

If the node $u_i$ appears in $C'$ at some position $j < i$, then removing that node is equivalent to replace the edges $u_{i-1}u_i$ and $u_iu_{i+1}$ by the edge $u_{i-1}u_{i+1}$, where $\ell(u_{i-1}u_i) + \ell(u_iu_{i+1}) \geq \ell(u_{i-1}u_{i+1})$. A cycle $C$ containing all nodes of $G$ is obtained by repeating the process and:

$$\sum_{e \in C} \ell(e) \leq \sum_{e \in T} \ell(e) + \sum_{e \in M} \ell(e) \leq \text{OPT} + \frac{1}{2} \times \text{OPT} = \frac{3}{2} \times \text{OPT}$$
Christofides algorithm for metric TSP

**Input:** \( G = (V, E) \) and \( \ell : E \rightarrow \mathbb{Q}_+ \).

1. \( T \leftarrow \) Minimum spanning tree of \( (G, \ell) \).

2. \( G' \leftarrow \) Induced subgraph of \( G \) considering only the nodes with odd degree in \( T \).

3. \( M \leftarrow \) Minimum perfect matching of \( (G', \ell) \).

4. \( C \leftarrow \) Eulerian circuit of \( T \cup M \).

5. \( C \leftarrow \) Remove repeated nodes of \( C \).

6. Return \( C \).
**Theorem.** The algorithm of Christofides is a polynomial time $\frac{3}{2}$-approximation algorithm for the metric TSP.
Primal method
The linear programming relaxation can be solved “efficiently” and its solutions may offer “good” bounds for the optimal values.

Hence, constructing integer solutions from a solution of the linear relaxation could guarantee some “quality” for the constructed solution.
The **minimum set cover** receives an instance \( I = \langle E, S, \omega \rangle \), where:

- \( E \) is a finite set of elements,
- \( S \) is a collection of subsets of \( E \), such that \( \bigcup_{S \in S} S = E \),
- \( \omega : S \rightarrow \mathbb{Q}_+ \) is a weight function over each set of \( S \).

The objective of the problem is to find a cover \( \mathcal{J} \subseteq S \), such that \( \bigcup_{S \in \mathcal{J}} S = E \) and \( \sum_{S \in \mathcal{J}} \omega(S) \) is minimized.
Define a binary variable $x_S$ for each $S \in S$, indicating if $S$ is selected or not to $\mathcal{T}$.

Model:

$$\min \sum_{S \in S} \omega(S) \times x_S$$

$$s.t.: \sum_{S | e \in S \in \mathcal{S}} x_S \geq 1 \quad \forall e \in E$$

$$x_S \in \{0, 1\} \quad \forall S \in S$$
Denote by $f_e$ the frequency of element $e \in E$:

$$f_e = |\{S | e \in S \in \mathcal{S}\}|.$$ 

The proposed algorithm follows:

1. $\bar{x} \leftarrow$ Linear relaxation solution of the model.

2. $f \leftarrow \max_{e \in E} f_e$.

3. $\mathcal{T} \leftarrow \left\{ S | S \in \mathcal{S} \land \bar{x}_S \geq \frac{1}{f} \right\}$.

4. Return $\mathcal{T}$. 
For each $e \in E$:

\[
\sum_{S \mid e \in S \in \mathcal{S}} x_S \geq 1
\]

\[
\Rightarrow \max_{S \mid e \in S \in \mathcal{S}} x_S \times f_e \geq 1
\]

\[
\Rightarrow \max_{S \mid e \in S \in \mathcal{S}} x_S \geq \frac{1}{f_e} \geq \frac{1}{f}
\]

Hence, \(\mathcal{T}\) contains at least one \(S \in \mathcal{S}\) such that \(e \in S\). Therefore:

\[
\bigcup_{S \in \mathcal{T}} S = E
\]
The selection of each $S \in \mathcal{T}$ guarantees $\bar{x}_S \geq \frac{1}{f}$. Thus:

$$\sum_{S \in \mathcal{T}} \omega(S) = f \times \sum_{S \in \mathcal{T}} \omega(S) \times \frac{1}{f} \leq f \times \sum_{S \in \mathcal{S}} \omega(S) \times \bar{x}_S$$

If $OPT$ is the optimal solution value, then, since $\bar{x}$ is a solution of the linear relaxation, $OPT \geq \sum_{S \in \mathcal{S}} \omega(S) \times \bar{x}_S$ and:

$$\sum_{S \in \mathcal{S}} \omega(S) \leq f \times OPT$$

**Theorem.** The proposed primal rounding technique is a polynomial time $f$-approximation algorithm for the minimum set cover.
Dual method
The linear programming relaxation of a dual linear program can be also solved “efficiently” and its solutions may offer “good” bounds for the optimal values.

Similarly to the primal method, the proposal is to construct an integer solution from a solution of the linear relaxation of a dual, proving the “quality” of the constructed solution.
Dual method

Dual formulation of minimum set cover

Primal: (relaxation)
\[
\min \sum_{S \in \mathcal{S}} \omega(S) \times x_S \\
\text{s.t.: } \sum_{S \mid e \in S \in \mathcal{S}} x_S \geq 1 \quad \forall e \in \mathcal{E} \\
x_S \geq 0 \quad \forall S \in \mathcal{S}
\]

Dual:
\[
\max \sum_{e \in \mathcal{E}} y_e \\
\text{s.t.: } \sum_{e \in S} y_e \leq \omega(S) \quad \forall S \in \mathcal{S} \\
y_e \geq 0 \quad \forall e \in \mathcal{E}
\]
Dual method

Rounding technique

1. \( \overline{y} \leftarrow \) Linear relaxation solution of the dual.

2. \( \mathcal{T} \leftarrow \left\{ S \mid S \in S \land \sum_{e \in S} \overline{y}_e = \omega(S) \right\} \).

3. Return \( \mathcal{T} \).
Feasibility of the algorithm solution

Suppose there exists $e' \in E$, such that $e' \notin \bigcup_{S \in \mathcal{S}} S$. Then, for each $S \in S$ with $e' \in S$: $\sum_{e \in S} y_e < \omega(S)$.

If $\epsilon = \min_{S \in S \land e' \in S} \left\{ \omega(S) - \sum_{e \in S} y_e \right\}$, then $y_e = \begin{cases} y_e + \epsilon, & e = e' \\ y_e, & e \neq e' \end{cases}$ satisfies:

$$\sum_{e \in S} y_e = \sum_{e \in S} y_e + \epsilon \leq \omega(S) \quad \forall S \in S \wedge e' \in S$$

$$\sum_{e \in S} y_e = \sum_{e \in S} y_e \leq \omega(S) \quad \forall S \in S \wedge e' \notin S$$

Thus, $y$ is feasible for the dual with value $\sum_{e \in E} y_e = \sum_{e \in E} y_e + \epsilon > \sum_{e \in E} y_e$, which contradicts the optimality of $y$. Therefore,

$$\bigcup_{S \in \mathcal{S}} S = E$$
The selection of each $S \in \mathcal{I}$ guarantees:

$$
\sum_{S \in \mathcal{I}} \omega(S) = \sum_{S \in \mathcal{I}} \sum_{e \in S} \bar{y}_e \leq \sum_{e \in E} f_e \times \bar{y}_e \leq f \times \sum_{e \in E} \bar{y}_e
$$

If $OPT$ is the optimal solution value, then, since $\bar{y}$ is a solution of the linear relaxation of the dual, $OPT \geq \sum_{e \in E} \bar{y}_e$ and:

$$
\sum_{S \in \mathcal{I}} \omega(S) \leq f \times OPT
$$

**Theorem.** The proposed dual rounding technique is a polynomial time $f$-approximation algorithm for the minimum set cover.
Primal-dual method
Consider primal and dual formulations for a problem, where $A \in \mathbb{R}^{m \times n}$:

**Primal:**

$$\begin{align*}
\text{min} & \quad c^t \times x \\
\text{s.t.} & \quad A \times x \geq b \\
& \quad x \geq 0
\end{align*}$$

**Dual:**

$$\begin{align*}
\text{max} & \quad b^t \times y \\
\text{s.t.} & \quad A^t \times y \leq c \\
& \quad y \geq 0
\end{align*}$$

Given $0 < \alpha \leq 1 \leq \beta$, a pair of (primal, dual) feasible solutions $(x', y')$ has **$\alpha$-approximated slacks on the primal** if for each $1 \leq j \leq n$:

$$x'_j = 0 \quad \text{or} \quad (A^t \times y')_j \geq \alpha \times c_j$$

The pair $(x', y')$ has **$\beta$-approximated slacks on the dual** if for each $1 \leq i \leq m$:

$$(A \times x')_i \leq \beta \times b_i \quad \text{or} \quad y'_i = 0$$
**Lemma.** If the (primal, dual) pair of feasible solutions \((x, y)\) satisfy the conditions for \(\alpha\)-approximated slacks on the primal and \(\beta\)-approximated slacks on the dual, then:

\[
\alpha \times c^t \times x \leq \beta \times b^t \times y
\]

**Proof:**

\[
\alpha \times c^t \times x \leq (A^t \times y)^t \times x = y^t \times (A \times x) \leq y^t \times (\beta \times b) = \beta \times b^t \times y
\]

\((\alpha\text{-approximated slacks property}) \quad \quad \quad \quad (\beta\text{-approximated slacks property})\)
Corollary. If the (primal, dual) pair of feasible solutions \((x, y)\) satisfy the conditions for \(\alpha\)-approximated slacks on the primal and \(\beta\)-approximated slacks on the dual, then:

\[
\begin{align*}
    c^t \times x \leq \frac{\beta}{\alpha} \times c^t \times x^* & \quad \text{and} \quad b^t \times y \geq \frac{\alpha}{\beta} \times b^t \times y^*
\end{align*}
\]

Where, \(x^*\) and \(y^*\) are, respectively, optimal solutions of the primal and the dual.
At each iteration consider a feasible solution \( y' \) of the dual and define the restricted primal approximated problem \( RPA(\alpha, \beta, A, b, c, y') \) and the restricted dual approximated problem \( RDA(\alpha, A, b, c, y') \) as follows:

\[
\begin{align*}
RPA(\alpha, \beta, A, b, c, y') & : \\
A \times x & \geq b \\
x & \geq 0 \\
(A \times x)_i & \leq \beta \times b_i, \quad \text{if } y'_i \neq 0 \\
x_i & = 0, \quad \text{if } (A^t \times y')_j < \alpha \times c_j
\end{align*}
\]

\[
\begin{align*}
RDA(\alpha, A, b, c, y') & : \\
b^t \times y & > 0 \\
(A^t \times y)_j & \leq 0, \quad \text{if } (A^t \times y')_j \geq \alpha \times c_j \\
y_i & \geq 0, \quad \text{if } y'_i = 0
\end{align*}
\]

If the \( RPA(\alpha, \beta, A, b, c, y') \) has a feasible solution \( \bar{x} \), then \( \alpha \times c^t \times \bar{x} \leq \beta \times b^t \times y' \) and \( c^t \times x \leq \frac{\beta}{\alpha} \times OPT \), where \( OPT \) is the optimal value of the primal.

Otherwise, there exists a feasible solution \( \bar{y} \) of the \( RDA(\alpha, A, b, c, y') \). Let \( \theta > 0 \) be the largest value such that \( y'' = y' + \theta \times \bar{y} \) is a feasible solution of the dual, and consider \( y'' \) for the next iteration.
Primal-dual method

Primal-dual approximation algorithm

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $0 < \alpha \leq 1 \leq \beta$.

1. $y \leftarrow$ a feasible solution of the dual.

2. While $RPA(\alpha, \beta, A, b, c, y)$ has not feasible solution:

3. $\bar{y} \leftarrow$ a feasible solution of $RDA(\alpha, A, b, c, y)$.

4. $\theta \leftarrow$ max positive such that $y + \theta \times \bar{y}$ is dual feasible.

5. $y \leftarrow y + \theta \times \bar{y}$.

6. Return a feasible solution of $RPA(\alpha, \beta, A, b, c, y)$.

Note. If $y + \theta \times \bar{y}$ is a feasible solution of the dual for any positive number $\theta$, then the dual is unlimited.
Primal-dual method

Primal and dual formulation for minimum set cover

Primal: (relaxation)
\[
\begin{align*}
\text{min} & \quad \sum_{S \in \mathcal{S}} \omega(S) \times x_S \\
\text{s.t.:} & \quad \sum_{S \mid e \in S \in \mathcal{S}} x_S \geq 1 \quad \forall e \in E \\
& \quad x_S \geq 0 \quad \forall S \in \mathcal{S}
\end{align*}
\]

Dual:
\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} y_e \\
\text{s.t.:} & \quad \sum_{e \in S} y_e \leq \omega(S) \quad \forall S \in \mathcal{S} \\
& \quad y_e \geq 0 \quad \forall e \in E
\end{align*}
\]
Primal-dual method

Approximation problems for minimum set cover

Let $\alpha = 1$, $\beta = f$ and $y'$ be a feasible solution of the dual, then:

\[ R_{PA}(E, S, \omega, y') : \]
\[
\sum_{S \in S \mid e \in S \in S} x_S \geq 1 \quad \forall e \in E
\]
\[
x \geq 0
\]
\[
\sum_{S \in S \mid e \in S \in S} x_S \leq f \quad \forall y'_e > 0
\]
\[
x_S = 0 \quad \forall S : \sum_{e \in S} y'_e < \omega(S)
\]

\[ R_{DA}(E, S, \omega, y') : \]
\[
\sum_{e \in E} y_e > 0
\]
\[
\sum_{e \in S} y_e \leq 0 \quad \forall S : \sum_{e \in S} y'_e = \omega(S)
\]
\[
y_e \geq 0 \quad \forall S : y'_e = 0
\]

There exists a feasible solution for $R_{PA}(E, S, \omega, y')$ iff for each $e' \in E$ there exists $S \in S$ such that $e' \in S$ and $\sum_{e \in S} y'_e = \omega(S)$.

If $y'$ guarantees the above property, then $x_S = \begin{cases} 1, & \sum_{e \in S} y'_e = \omega(S) \\ 0, & \sum_{e \in S} y'_e < \omega(S) \end{cases}$ is a feasible solution and an $f$-approximation for the optimal value.
**Primal-dual method**

**Primal-dual approximation algorithm**

**Input:** $E$, $S$ and $\omega : S \rightarrow \mathbb{Q}_+$.

1. $y \leftarrow 0_{|E|}$.
2. While $\exists \bar{e} : \forall S \in S \land \bar{e} \in S, \sum_{e \in S} y_e < \omega(S)$:
   3. $\bar{y} \leftarrow$ unitary vector, where $\bar{y}_{\bar{e}} = 1$.
   4. $\theta \leftarrow \min_{S \in S \land \bar{e} \in S} \left\{ \omega(S) - \sum_{e \in S} y_e \right\}$.
   5. $y \leftarrow y + \theta \times \bar{y}$.
6. Return $\left\{ S | S \in S \land \sum_{e \in S} y_e = \omega(S) \right\}$.

At each iteration a new element is covered, thus the maximum number of iterations is $\mathcal{O}(|E|)$. 
Approximability Classes
Approximability classes

For a minimization problem $\Pi$ an algorithm $A$ is a full polynomial time approximation scheme (\textit{FPTAS}), if there exists a polynomial function $p$, such that for any instance $I$ and any $\epsilon > 0$:

- the time complexity of $A$ is $\mathcal{O}(p(I, \frac{1}{\epsilon}))$, and

$\text{val}(A(I)) \leq (1 + \epsilon) \times opt_{\Pi}(I)$

\textit{FPTAS} is the class of optimization problems for whom there exists a full polynomial time approximation scheme.

For maximization problems the inequality is $\text{val}(A(I)) \geq (1 - \epsilon) \times opt_{\Pi}(I)$
For a minimization problem $\Pi$ an algorithm $A$ is a polynomial time approximation scheme (PTAS), if for each $\epsilon > 0$ there exists a polynomial function $p$, such that for any instance $I$:

- the time complexity of $A$ is $O(p(I))$, and

- $val(A(I)) \leq (1 + \epsilon) \times opt_{\Pi}(I)$

PTAS is the class of optimization problems for whom there exists a polynomial time approximation scheme.

For maximization problems the inequality is $val(A(I)) \geq (1 - \epsilon) \times opt_{\Pi}(I)$
Approximability classes

For a minimization problem $\Pi$ and a constant $\alpha > 0$, an algorithm $A$ is a **polynomial time $\alpha$-approximation**, if there exists a polynomial function $p$, such that for any instance $I$:

- the time complexity of $A$ is $\mathcal{O}(p(I))$, and
- $val(A(I)) \leq \alpha \times opt_{\Pi}(I)$

$APX$ is the class of optimization problems for whom there exists a polynomial time $\alpha$-approximation, for some constant $\alpha > 0$.

For maximization problems the inequality is $val(A(I)) \geq \alpha \times opt_{\Pi}(I)$.
Approximability classes

Diagram of classes

FPTAS  PTAS  APX  NPO
Unless $P = NP$:

$$PO \subset FPTAS \subset PTAS \subset APX \subset NPO$$

Similar to $NP$-hard and $NP$-complete definitions, each class has associated hard and complete classes of problems:

- **Hard.** A problem $\Pi$ is hard for a class $\mathcal{C}$ iff each problem $\Pi' \in \mathcal{C}$ can be reduced to $\Pi$.

- **Complete.** A problem $\Pi$ is complete for a class $\mathcal{C}$ iff $\Pi$ is hard for $\mathcal{C}$ and $\Pi \in \mathcal{C}$.

In this context, the reductions must be described by polynomial time algorithms that preserve the approximation ratio. Example of these reductions are: $P$-reduction, $L$-reduction, $E$-reduction.
An inapproximability result for a problem $\Pi$ is a proof showing that $\Pi$ cannot be $\alpha$-approximated in polynomial time, unless $P = NP$. 
Suppose there exists a polynomial time $\alpha$-approximation algorithm $A$ for TSP on general graphs, where $\alpha$ is a constant value.

Consider an instance $G = (V, E)$ of the Hamiltonian cycle problem which is $NP$-complete. This problem receives an undirected graph and asks for the existence of a cycle passing through all nodes.

Construct an instance of the TSP with the same set of nodes $V$, where

$$\ell(uv) = \begin{cases} 1, & uv \in E \\ (\alpha + 1) \times |V|, & uv \notin E \end{cases}.$$ 

If there exist a Hamiltonian cycle in $G$, then the optimal solution value of the TSP is $|V|$ and the $A$ must find a solution whose value is at most $\alpha \times |V|$. Thus, the solution found by $A$ has no edge with length $(\alpha + 1) \times |V|$. Therefore, such solution is a Hamiltonian cycle, implying that the Hamiltonian cycle problem can be solved in polynomial time.

**Theorem.** Unless $P = NP$, TSP on general graphs does not admit a polynomial time $\alpha$-approximation algorithm, where $\alpha$ is a constant value.
Approximation Algorithms

Integer linear programming: formulations, techniques and applications.

Santiago Valdés Ravelo
santiago.ravelo@inf.ufrgs.br

18 November, 2020