Feasibility Pump Heuristics for Column Generation Approaches

Ruslan Sadykov[2]
Pierre Pesneau[1,2] Francois Vanderbeck[1,2]

[1]University Bordeaux I

[2]INRIA Bordeaux — Sud-Ouest

SEA 2012
Bordeaux, France, June 9, 2012

Outline

Generic Primal Heuristics for MIPs

“good” feasible solutions using the tools of **exact optimization**

Generic Primal Heuristics for MIPs

"good" feasible solutions using the tools of **exact optimization**

- Truncating an exact method
- Building from the relaxation used for the exact approach
- Defining a target based on the relaxation
- Using dual information to price choices in greedy heuristics
- Exact approach used to explore a neighborhood

Generic Primal Heuristics for MIPs

"good" feasible solutions using the tools of **exact optimization**

- Truncating an exact method
- Building from the relaxation used for the exact approach
- Defining a target based on the relaxation
- Using dual information to price choices in greedy heuristics
- Exact approach used to explore a neighborhood

Examples: **[Berthold'06]**

1. *Large Scale Neighborhood Search* **[Ahuja al'02]**
2. *Relaxation Induced Neighborhood Search* **[Dana al'05]**
3. *Local Branching* **[Fischetti al'03]**
4. *Feasibility Pump* **[Fischetti al'05]**

Generic LP based heuristics

$$\min\{\textstyle\sum_j c_j x_j : \sum_j a_{ij} x_j \geq b_i \,\forall i, \;\; l_j \leq x_j \leq u_j \,\forall j\}$$

Generic LP based heuristics

$$\min\{\sum_j c_j x_j : \sum_j a_{ij} x_j \geq b_i \, \forall i, \;\; l_j \leq x_j \leq u_j \, \forall j\}$$

- Rounding: Iteratively **select** a var $\boldsymbol{x_j}$ and **bound**/fix **it**
  - *least fractional:* $\mathbf{argmin}_j\{\mathbf{min}\{\boldsymbol{x_j} - \lfloor \boldsymbol{x_j} \rfloor, \lceil \boldsymbol{x_j} \rceil - \boldsymbol{x_j}\}\}$
  - *guided search:* $\mathbf{argmin}_j\{|\boldsymbol{x_j} - \boldsymbol{x_j}^{\mathbf{inc}}|\}$

Generic LP based heuristics

$$\min\{\sum_j c_j x_j : \sum_j a_{ij} x_j \geq b_i \; \forall i, \;\; l_j \leq x_j \leq u_j \; \forall j\}$$

- Rounding: Iteratively **select** a var $\boldsymbol{x_j}$ and **bound**/fix **it**
  - *least fractional:* $\mathbf{argmin}_j\{\mathbf{min}\{\boldsymbol{x_j} - \lfloor \boldsymbol{x_j} \rfloor, \lceil \boldsymbol{x_j} \rceil - \boldsymbol{x_j}\}\}$
  - *guided search:* $\mathbf{argmin}_j\{|\boldsymbol{x_j} - \boldsymbol{x_j^{\mathbf{inc}}}|\}$
- Diving: rounding + LP resolve + reiterate
  heuristic depth search in branch-and-bound tree
  branching rule $\neq$ that of exact branch-and-bound

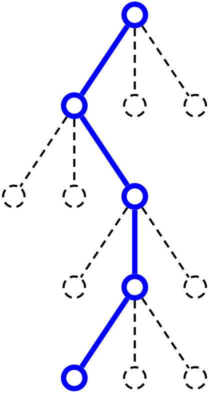
Generic LP based heuristics

$$\min\{\sum_j c_j x_j : \sum_j a_{ij} x_j \geq b_i \,\forall i, \;\; l_j \leq x_j \leq u_j \,\forall j\}$$

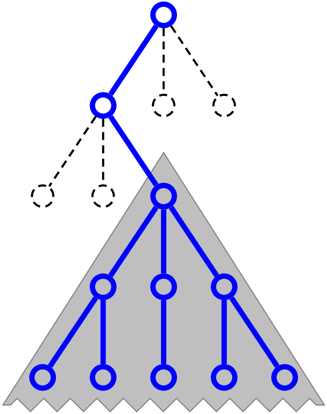
- Rounding: Iteratively **select** a var $\boldsymbol{x_j}$ and **bound**/fix **it**
  - *least fractional:* $\mathbf{argmin}_j\{\mathbf{min}\{\boldsymbol{x_j} - \lfloor \boldsymbol{x_j} \rfloor, \lceil \boldsymbol{x_j} \rceil - \boldsymbol{x_j}\}\}$
  - *guided search:* $\mathbf{argmin}_j\{|\boldsymbol{x_j} - \boldsymbol{x_j^{\mathbf{inc}}}|\}$
- Diving: rounding + LP resolve + reiterate
  heuristic depth search in branch-and-bound tree
  branching rule $\neq$ that of exact branch-and-bound
- sub-MIPing: rounding/diving + MIP sol of the residual prob.

Heuristic search in branch-and-bound tree

Diving



sub-MIPing



Feasibility Pump heuristic

Target solution $\tilde{x}$ is obtained by rounding LP solution x^{LP} to the closest integer solution. If $\tilde{x}$ is not feasible, the problem is modified:

- $0 - 1$ integer program

$$\min\left\{c\,x + \epsilon\Big(\sum_{j:\ \tilde{x}_j=0} x_j + \sum_{j:\ \tilde{x}_j=1} (1 - x_j)\Big) : A\,x \geq a,\ x \in [0,1]^n\right\}$$

- general integer program ($l_j \leq x_j \leq u_j$)

$$\min\Big\{c\,x + \epsilon\Big(\sum_{j:\tilde{x}_j=l_j} (x_j - l_j) + \sum_{j:\tilde{x}_j=u_j} (u_j - x_j) + \sum_{j:l_j<\tilde{x}_j<u_j} d_j\Big) : A\,x \geq a,$$
$$d_j - \tilde{x}_j \leq x_j \leq d_j + \tilde{x}_j\ \forall j,\ x \in \mathbb{R}^n\Big\}$$

The Branch-and-Price Approach

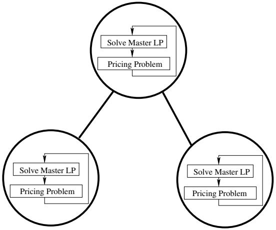
$$
\begin{array}{lccccccccl}
\min & c^1 x^1 & + & c^2\, x^2 & + & \dots & + & c^K\, x^K & & \\
 & D\, x^1 & + & D\, x^2 & + & \dots & + & D\, x^K & \geq d \\
 & B\, x^1 & & & & & & & \geq b \\
 & & & B\, x^2 & & & & & \geq b \\
 & & & & & \ddots & & & \geq \vdots \\
 & & & & & & & B\, x^K & \geq b \\
 & x^1 & \in \mathbb{N}^n, & x^2 & \in \mathbb{N}^n, & \dots & x^K & \in \mathbb{N}^n. &
\end{array}
$$

The Branch-and-Price Approach

$$
\begin{array}{lccccccccl}
\min & c^1 x^1 & + & c^2 x^2 & + & \dots & + & c^K x^K & & \\
 & D x^1 & + & D x^2 & + & \dots & + & D x^K & \geq d \\
 & B x^1 & & & & & & & \geq b \\
 & & & B x^2 & & & & & \geq b \\
 & & & & & \ddots & & & \geq \vdots \\
 & & & & & & & B x^K & \geq b \\
 & x^1 & \in \mathbb{N}^n, & x^2 & \in \mathbb{N}^n, & \dots & x^K & \in \mathbb{N}^n. &
\end{array}
$$

Relax $Dx \geq d \Rightarrow$ decomposition: **subproblem** $\{B x \geq b, x \in \mathbb{N}^n\}$ and a **reformulation** solved by **Branch-and-Price**:

$$
\begin{array}{rcl}
\min \sum_{g \in G} c x^g \lambda_g & & \\
\sum_{g \in G} D x^g \lambda_g & \geq & d \\
\sum_{g \in G} \lambda_g & = & K \\
\lambda & \in & \mathbb{N}^{|G|}
\end{array}
$$



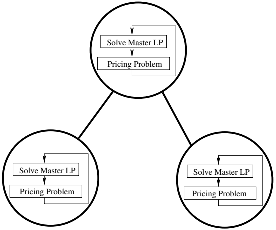
The Branch-and-Price Approach

$$
\begin{array}{llllllllll}
\min & c^1 x^1 & + & c^2\, x^2 & + & \dots & + & c^K\, x^K & \\
 & D\, x^1 & + & D\, x^2 & + & \dots & + & D\, x^K & \geq d \\
 & B\, x^1 & & & & & & & \geq b \\
 & & & B\, x^2 & & & & & \geq b \\
 & & & & & \ddots & & & \geq \vdots \\
 & & & & & & & B\, x^K & \geq b \\
 & x^1 & \in \mathbb{N}^n, & x^2 & \in \mathbb{N}^n, & \dots & x^K & \in \mathbb{N}^n. &
\end{array}
$$

Relax $Dx \geq d \Rightarrow$ decomposition: **subproblem** $\{B\,x \geq b, x \in \mathbb{N}^n\}$ and a **reformulation** solved by **Branch-and-Price**:

$$
\begin{array}{rcl}
\min \sum_{g \in G} c x^g \ \lambda_g & & \\
\sum_{g \in G} D x^g \ \lambda_g & \geq & d \\
\sum_{g \in G} \lambda_g & = & K \\
\lambda & \in & \mathbb{N}^{|G|}
\end{array}
$$

$$
y := \sum_k x^k = \sum_{g \in G} x^g \ \lambda_g
$$



Difficulties in a B-a-P context for generic heuristics

Heuristic paradigm in original space or the reformulation?

Difficulties in a B-a-P context for generic heuristics

Heuristic paradigm in original space or the reformulation?

On master variables: λ (aggregated decisions)

- **Cannot** fix bounds (as in rounding)
- **Cannot** modify costs (as in feasibility pump)

Difficulties in a B-a-P context for generic heuristics

Heuristic paradigm in original space or the reformulation?

On master variables: λ (aggregated decisions)

- **Cannot** fix bounds (as in rounding)
- **Cannot** modify costs (as in feasibility pump)

On original variables: x (disaggregated decisions)

- **Cannot** grasp individual SP var. after aggregation in the common case of identical SPs
- **Cannot** modify the SP structure required by the oracle

Difficulties in a B-a-P context for generic heuristics

Heuristic paradigm in original space or the reformulation?

On master variables: λ (aggregated decisions)

- **Cannot** fix bounds (as in rounding)
- **Cannot** modify costs (as in feasibility pump)

On original variables: x (disaggregated decisions)

- **Cannot** grasp individual SP var. after aggregation in the common case of identical SPs
- **Cannot** modify the SP structure required by the oracle

Differences

- Acting on master λ variables results in a **more macroscopic decision.**
- Faster progress to an integer solution, but you can quickly "**paint yourself in a corner**"

Generic modifications of the master

- Setting a lower bound of a column: $\lambda_g \geq l_g$
- Decreasing cost c_g of a column λ_g

Generic modifications of the master

- Setting a lower bound of a column: $\lambda_g \geq l_g$
- Decreasing cost c_g of a column λ_g

In both cases, pricing oracle overestimates the reduced cost of column λ_g already included in the master.

Generic modifications of the master

- Setting a lower bound of a column: $\lambda_g \geq l_g$
- Decreasing cost c_g of a column λ_g

In both cases, pricing oracle overestimates the reduced cost of column λ_g already included in the master.

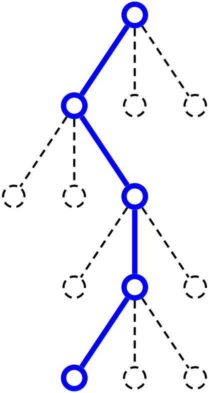
Preprocessing

- Lower bound setting is done by fixing a partial ("rounded-down") solution
- After that, the residual master problem is defined by preprocessing:
  - updating RHS of the master;
  - updating bounds for subproblem variables.

Pure Diving Heuristic

Depth-First Search

- select *least fractional* col: $\lambda_s \leftarrow \lceil \bar{\lambda}_s \rfloor$
- update master and SP
- apply preprocessing



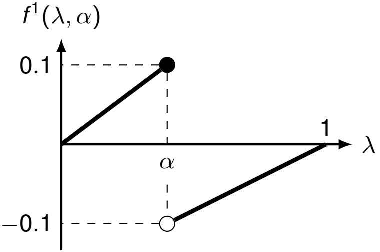
Generic Feasibility Pump algorithm I

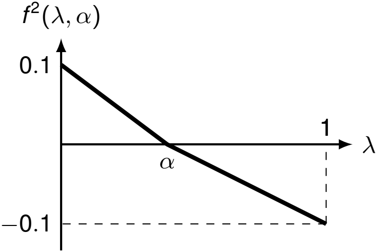
- Solution $\tilde{\lambda}$ is defined by rounding the LP solution λ^{LP}.
- If $\tilde{\lambda}$ is feasible, stop. Otherwise, we use $\tilde{\lambda}$ as a target point.
- We decrease the cost of rounded-up columns and increase the cost of rounded-down ones (but not beyond the original cost).

Generic Feasibility Pump algorithm I

- Solution $\tilde{\lambda}$ is defined by rounding the LP solution λ^{LP}.
- If $\tilde{\lambda}$ is feasible, stop. Otherwise, we use $\tilde{\lambda}$ as a target point.
- We decrease the cost of rounded-up columns and increase the cost of rounded-down ones (but not beyond the original cost).

Cost modification factor functions





$$f^1(\lambda, \alpha) = \begin{cases} 0.1 \quad \frac{\lambda}{\alpha} & \text{if } \lambda \leq \alpha \\ -0.1 \quad \frac{(1-\lambda)}{(1-\alpha)} & \text{if } \lambda > \alpha \end{cases}$$

$$f^2(\lambda, \alpha) = \begin{cases} 0.1 \quad (1 - \frac{\lambda}{\alpha}) & \text{if } \lambda \leq \alpha \\ -0.1 \quad \frac{(\lambda-\alpha)}{(1-\alpha)} & \text{if } \lambda > \alpha \end{cases}$$

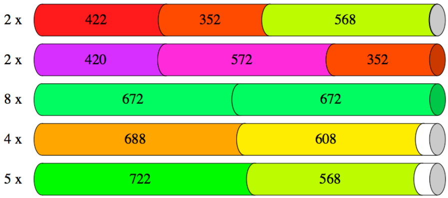
Embedding Feasibility Pump in a Diving heuristic

- At iteration t, the modified master becomes

$$\min \Big\{ \sum_{g \in G^t} c_g^t \lambda_g : \sum_{g \in G^t} (Ax^g)\lambda_g \geq a^t; \ \sum_{g \in G^t} \lambda_g = K^t; \ \lambda_g \in \mathbb{N} \quad \forall g \in G$$

- Before defining target solution $\tilde{\lambda}^t$, the "rounded-down" integer part of λ_{LP}^t is fixed and removed: $\lambda_g^t \leftarrow \lambda_g^t - \lfloor \lambda_g^t \rfloor$ (this way the residual master is close to a $0-1$ problem).
- Cycling can occur if no columns are rounded up in $\tilde{\lambda}^t$. In this case, we decrease fractionality threshold parameter α (initially $\alpha \leftarrow 0.5$).

Cutting Stock Problem

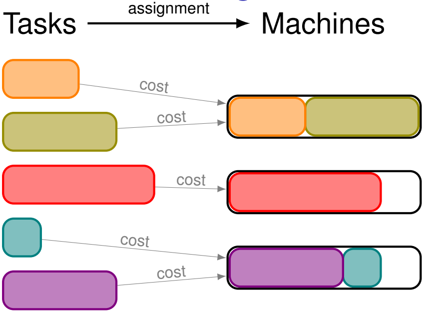


$$
\begin{aligned}
n &= 50, 100 \\
d_i &\in [1, 50] \\
W &= 10000 \\
w_i &\in [500, 2500]
\end{aligned}
$$

50 instances for each n

n	function	found	opt	gap	time
50	Pure Div.	50/50	43/50	0.07	1.17
50	f^1	50/50	45/50	0.05	6.14
50	f^2	50/50	41/50	0.09	4.82
100	Pure Div.	50/50	35/50	0.08	4.08
100	f^1	50/50	43/50	0.04	23.93
100	f^2	50/50	40/50	0.05	17.98

Generalized Assignment



Instances from OR-Library (type D)

50 instances for each (n, m)

m	n	function	found	gap	time
10	50	Pur Div.	34/50	1.00%	0.37
10	50	f^1	36/50	0.98%	1.81
10	50	f^2	48/50	1.14%	0.81
20	100	Pur Div.	35/50	0.65%	2.46
20	100	f^1	36/50	0.55%	14.56
20	100	f^2	42/50	0.75%	5.92

Conclusions

Summary

- Feasibility Pump heuristic can be extended to the column generation context
- The key is to restrict problem modifications to setting lower bound and cost reduction.
- Compared with the generic diving heuristic, feasibility pump heuristic produced more feasible primal solutions without loosing on the quality.

Conclusions

Summary

- Feasibility Pump heuristic can be extended to the column generation context
- The key is to restrict problem modifications to setting lower bound and cost reduction.
- Compared with the generic diving heuristic, feasibility pump heuristic produced more feasible primal solutions without loosing on the quality.

Future work

- Adaptation of diversification mechanisms for the Feasibility Pump heuristic
- Numerical tests on a larger scope of applications
- Compare Feasibility Pump heuristic on aggregated variables λ versus Feasibility Pump in the space of original variables x.