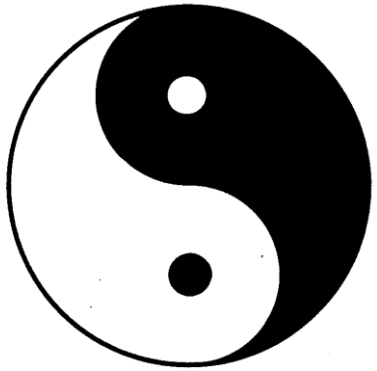
Hybrid Approaches for 0-1 Mixed Integer Program

Said.hanafi@univ-valenciennes.fr

Ecole Polytechnique

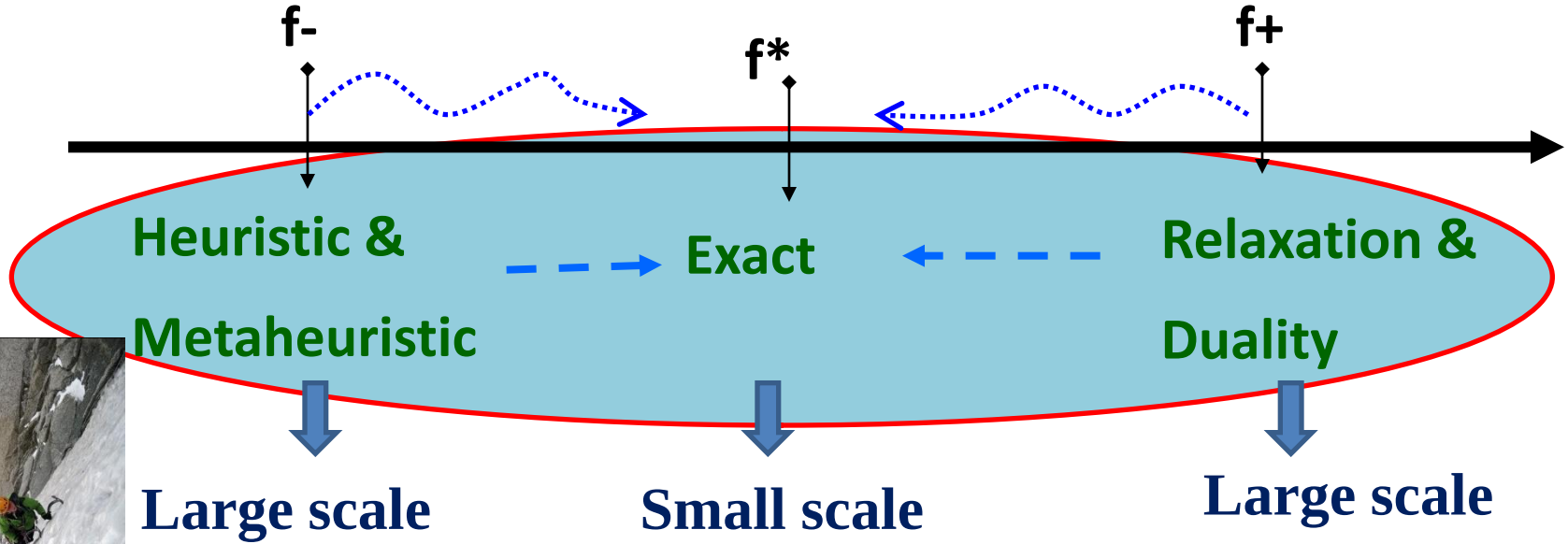
15 Janvier 2013



Hybrid Approaches

- **Hard Optimization Problem**

$$f^* = \text{Max}\{f(x) : x \in X \subseteq E\}$$



Google ROADEF/EURO challenge 2011-2012
Team S21

Hybrid Method for Machine Reassignment Problem

Saïd Hanafi	University of Valenciennes, France
Hideki Hashimoto	Nagoya University, Japan
Koji Nonobe	Hosei University, Japan
Michel Vasquez	Ales School of Mines, France
Yannick Vimont	Ales School of Mines, France
Mutsunori Yagiura	Nagoya University, Japan

Size of MIP

	m	*p*	#0-1var	#var	#cont
b_1	100	5 000	525 120	1 201	300 504
b_2	100	5 000	524 620	1 301	294 354
b_3	100	20 000	2 150 250	601	1 778 725
b_4	500	20 000	10 086 600	3 501	988 224
b_5	100	40 000	4 350 820	601	4 145 394
b_6	200	40 000	8 734 000	1 401	3 814 160
b_7	4 000	40 000	160 752 500	28 001	61 125 850
b_8	100	50 000	5 450 300	301	5 318 910
b_9	1 000	50 000	50 460 900	4 001	5 156 163
b_10	5 000	50 000	250 489 600	20 001	25 073 872

An example

Instance	Lower Bound	Upper Bound
A2-1	65	391 189 190

Some Contributions with Hybridization

Exact Methods

Branch-and-Bound

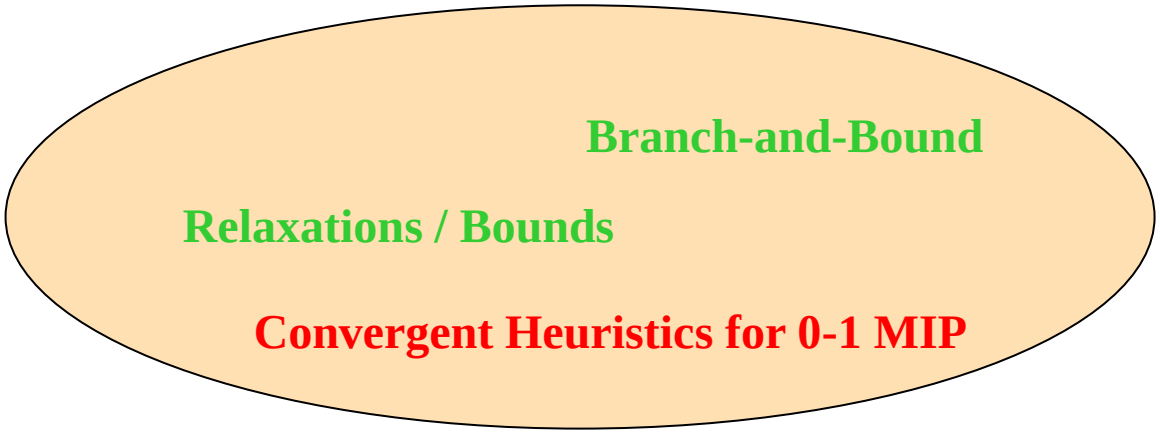
Relaxations / Bounds

Dynamic Programming

Heuristics / Metaheuristics

Scatter Search

Tabu Search
Oscillation Strategy
Variable Neighbourhood Se



Other techniques

- **Decomposition / Projection**
- **Fixation: Temporary or definitvely**
- **Cuts / Pseudo-Cuts**

Adaptive Memory
Intensification & Diversification

Some Contributions with Hybridization

Exact Methods

Branch-and-Bound

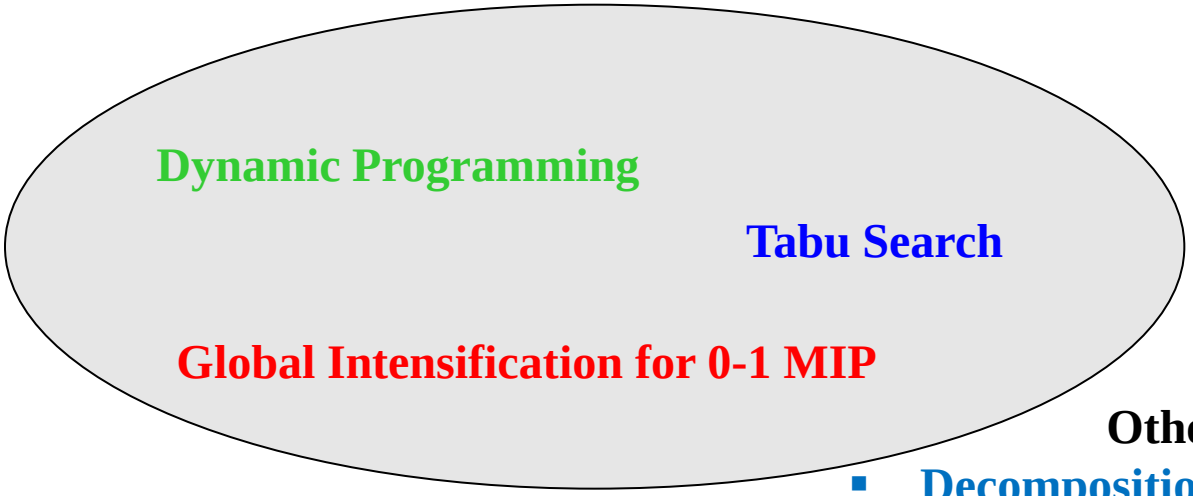
Relaxations / Bounds

Dynamic Programming

Heuristics / Metaheuristics

Scatter Search

Tabu Search
Oscillation Strategy
Variable Neighbourhood Se



Other techniques

- **Decomposition / Projection**
- **Fixation: Temporary or definitvely**
- **Cuts / Pseudo-Cuts**

Adaptive Memory
Intensification & Diversification

Some Contributions with Hybridization

Exact Methods

Branch-and-Bound

Relaxations / Bounds

Dynamic Programming

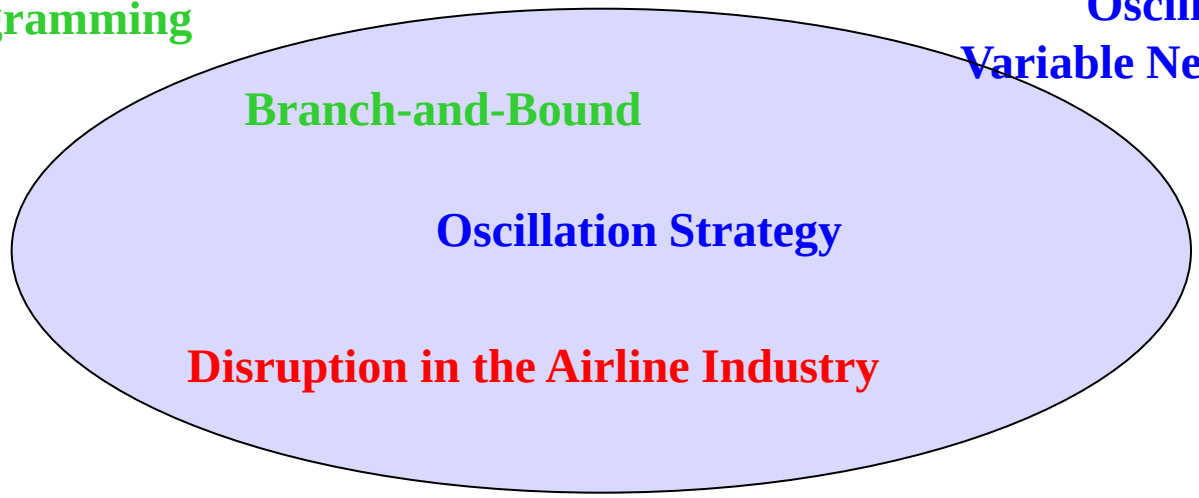
Heuristics / Metaheuristics

Scatter Search

Tabu Search

Oscillation Strategy

Variable Neighbourhood Se



Other techniques

- **Decomposition / Projection**
- **Fixation: Temporary or definitvely**
- **Cuts / Pseudo-Cuts**

Adaptive Memory

Intensification & Diversification

Some Contributions with Hybridization

Exact Methods

Branch-and-Bound

Relaxations / Bounds

Dynamic Programming

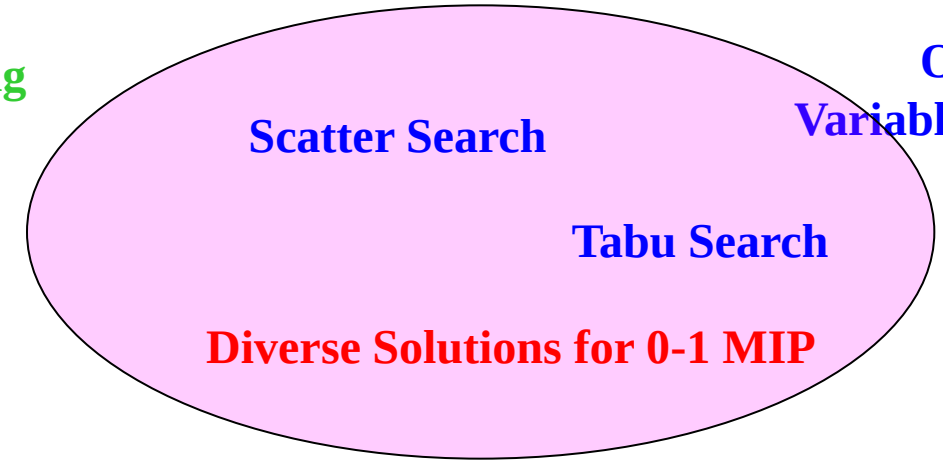
Heuristics / Metaheuristics

Scatter Search

Tabu Search

Oscillation Strategy

Variable Neighbourhood Se



Other techniques

- **Decomposition / Projection**
- **Fixation: Temporary or definitvely**
- **Cuts / Pseudo-Cuts**

Adaptive Memory

Intensification & Diversification

Some Contributions with Hybridization

Exact Methods

Branch-and-Bound

Relaxations / Bounds

Dynamic Programming

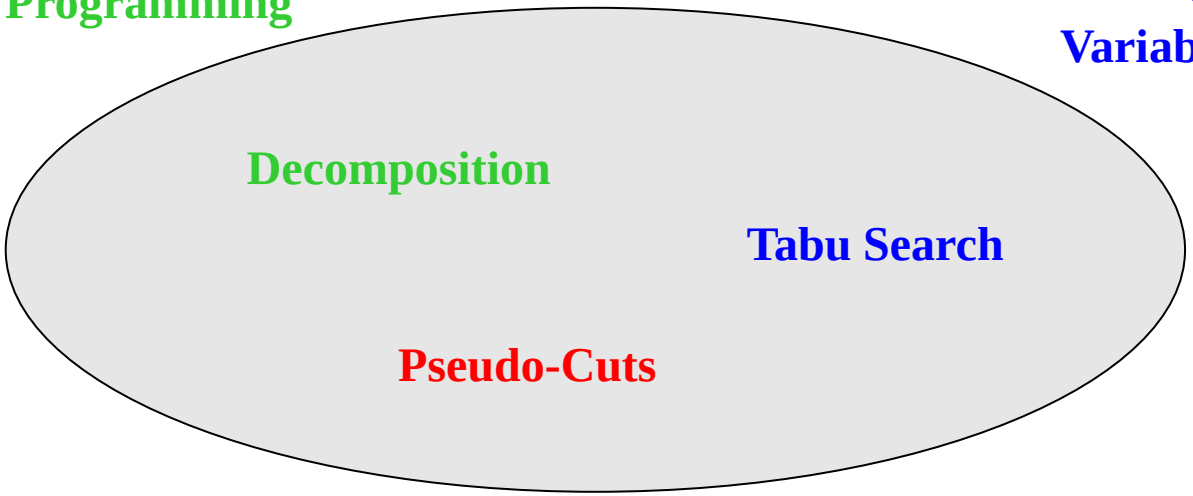
Heuristics / Metaheuristics

Scatter Search

Tabu Search

Oscillation Strategy

Variable Neighbourhood Se



Other techniques

- **Decomposition / Projection**
- **Fixation: Temporary or definitvely**
- **Cuts / Pseudo-Cuts**

Adaptive Memory

Intensification & Diversification

Outline

- MIP & Variants of Knapsack problem
- Relaxations of MIP
- Iterative Relaxation-based Heuristics (IRH)
- Hybrid VNDS with IRH
- Tabu Search & Dynamic Programming
- Tabu Search with Relaxation & Decomposition
- Conclusions

Mixed Integer Program

$$(MIP)\begin{cases} \texttt{maximiser} & cx \\ \texttt{s.c.} & Ax \leq b \\ & 0 \leq x_j \leq u_j \quad j = 1, \ldots, n \\ & x_j \ \texttt{entier} \quad j = 1, \ldots, p \end{cases}$$

n : nombre de variables
p : nombre de variables discrètes
m : nombre de contraintes

c(*n*), *A*(*m*,*n*), *b*(*m*)

- Programmation linéaire **(*PL*)** **$p = 0$**
- Progr. en variables entières **(*IP*)** **$p = n$**
- Progr. en var. 0-1 mixtes **(0-1*MIP*)** **$u_j = 1 \quad j = 1, \ldots, p$**
- Progr. en var. 0-1 **(0-1*IP*)** **$u_j = 1 \quad j = 1, \ldots, p \quad n = p$**
- Sac-à-dos multidimensionnel **(*SADM*)** **$u_j = 1 \quad j = 1, \ldots, p \quad n = p$**

$A, b, c \geq 0$

Multidimensional Knapsack Problem

$$(MKP) \quad \begin{cases} \max \ cx \\ s.t. \ Ax \le b \\ x \in \{0,1\}^n \end{cases} \qquad A(m,n), b(m), c(n) \ge 0$$

MKP* is *NP-Hard

- **Dantzig 1957, Lorie & Savage 1955, Markowitz & Manne 1957**
- **Martello & Toth, 1990; Kellerer & Pferschy & Pisinger 2004**
- **Fréville & Hanafi 2005, Hanafi & Wilbaut 2009**
- **Boussier & Vasquez & Vimont & Hanafi & Michelon 2009**
- **Hanafi & Lazic & Mladenovic & Wilbaut 2009**

Applications

- **Resource allocation problems**
- **Production scheduling problems**
- **Transportation management problems**
- **Adaptive multimedia system with QoS**
- **Telecommunication networks problems**
- **Wireless switch design**
- **Service selection for web services with QoS constraints**
- **Polyedral approach, (1,*k*)-configuration**

- **Sub-problem : Lagrangean / Surrogate relaxation**
- **Benchmark**

Variant of Knapsack Problem

- **Unidimensional Knapsack Problem** → $m = 1$
- **Bi-Dimensional Knapsack Problem** → $m = 2$

Objective

- *Quadratic*
- *Hyperbolic*
- *Min-Max*
- *Multi-Objectives*
- *Bi-Level*

Constraints

- *Cardinality*
- *Multiple-choice*
- *Precedence*
- *Demand*
- *Quadratic*

Multidemand Multidimensional Knapsack Problem

Cappanera & Trubian 2004

Knapsack & Covering Problem

$$(MDMKP)\begin{cases} \texttt{maximize} & \sum_{j=1}^{n} c_j x_j \\ \texttt{subject to} & \sum_{i=1}^{n} a_{1j}^{i} x_j \leq b_{1i} \quad \forall i = 1, \ldots, m_1 \\ & \sum_{i=1}^{n} a_{2j}^{i} x_j \geq b_{2i} \quad \forall i = 1, \ldots, m_2 \\ & x_j \in \{0;1\} \qquad j = 1, \ldots, n \end{cases}$$

MDMKP is a generalization of MKP

Multi-Choice Multidimensional Knapsack Problem

Nauss 78, Sinha & Zoltners 79

$$(MMKP)\quad \begin{cases} \max & \displaystyle\sum_{i=1}^{g}\sum_{j=1}^{g_i} c_{ij}x_{ij} & \\ s.t. & \displaystyle\sum_{i=1}^{g}\sum_{j=1}^{g_i} a_{ij}^{k}x_{ij} \le b^{k} & k=1,\ldots,m \\ & \displaystyle\sum_{j=1}^{g_i} x_{ij} = 1 & i=1,\ldots,g \\ & x_{ij} \in \{0,1\} & i=1,\ldots,g,\ j=1,\ldots,g_i \end{cases}$$

- g : **Number of groups**
- g_i : **Number of items in group i**
- c_{ij} : **Profit associated to item j of group i**
- a_{ij}^{k} : **Consuming resource k by item j of group i**
- b^{k} : **Capacity of resource k**

Multidimensional Knapsack Problem with Generalized Upper Bound constraints

$$(GUBMKP)\quad \begin{cases} \max & \displaystyle\sum_{i=1}^{g}\sum_{j=1}^{g_i} c_{ij}x_{ij} & \\ s.t. & \displaystyle\sum_{i=1}^{g}\sum_{j=1}^{g_i} a_{ij}^{k}x_{ij} \le b^{k} & k=1,\ldots,m \\ & \displaystyle\sum_{j=1}^{g_i} x_{ij} \le 1 & i=1,\ldots,g \\ & x_{ij} \in \{0,1\} & i=1,\ldots,g,\ j=1,\ldots,g_i \end{cases}$$

GUBMKP is a relaxation of MMKP

Knapsack Constrained Maximum Spanning Tree Problem

Aggarwal et al (1982)

$$(KCMST)\quad \begin{cases} \max & \sum_{a\in A} c_a x_a \\ s.t. & \sum_{a\in A} w_a x_a \le b \\ & x \text{ covering tree of } G \\ & x_a \in \{0,1\} \qquad a \in A \end{cases}$$

KCMST problem arises in practice in situations where the aim is to design a mobile communication network under a strict limit on total costs

Generalized Knapsack Problem

Suzuki (1978)

$$(GKP)\quad \begin{cases} \max & \sum_{j=1}^{n} c_j x_j \\ s.t. & \sum_{j=1}^{n} x_j \le c \\ & a_j y_j \le x_j \le b_j y_j \qquad j = 1,\ldots,n \\ & x_j \ge 0; y_j \in \{0,1\} \qquad j = 1,\ldots,n \end{cases}$$

Disjunctively Constrained Knapsack Problem

Yamada et al (2002)

$$(DCKP)\begin{cases} \max & \sum_{j=1}^{n} c_j x_j \\ s.t. & \sum_{j=1}^{n} a_j x_j \le b \\ & x_i + x_j \le 1 \qquad (i,j) \in I \\ & x_j \in \{0,1\} \qquad j = 1,\ldots,n \end{cases}$$

Max Min Multidimensional Knapsack Problem

$$(MMMKP)\begin{cases} Max \;\; Min\left\{c^k x : k = 1,\ldots, p\right\} \\ \quad s.t. \\ \quad\quad Ax \le b \quad x \in \{0,1\}^n \end{cases}$$

Introduction to minimax, Demyanov Molozemov 1974

Minimax and applications, Du, Pardalos 1995

Robust optimization applications, Yu 1996

A virtual pegging approach, Taniguchi, Yamada, Kataoka 2009

Iterative Heuristics, Hanafi, Mansi, Wilbaut 2012

Knapsack Sharing Problem

Brown (1979)

$$(KSP)\quad \begin{cases} \max \quad \min\left\{\sum_{j\in N_i} c_j x_j : i = 1,\ldots,m\right\} & \\ s.t. \quad \sum_{j\in N} a_j x_j \le b & \\ N = \bigcup_{i=1}^{m} N_i, N_i \cap N_l = \Phi & i \ne l \\ x_j \in \{0,1\} & j \in N \end{cases}$$

KSP is a generalization of KP

Knapsack Problem with Setup

Ham et al. (1985)

$$(KPS)\begin{cases} Max \ \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}x_{ij} + \sum_{i=1}^{N} f_i y_i \\ \text{s.t} \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} a_{ij}x_{ij} + \sum_{i=1}^{N} d_i y_i \leq b \\ x_{ij} \leq y_i, j = 1, \dots, n_i ; i = 1, \dots, N \\ x_{ij}, y_i \in \{0,1\}, j = 1, \dots, n_i ; i = 1, \dots, N \end{cases}$$

- N is the number of families
- n_i is the number of jobs in family i
- c_{ij} is the profit of job j in family i
- a_{ij} is the time to process job j in family i
- f_i is the setup cost for family i $(f_i < 0)$
- d_i is the setup time for family i
- b is the time available for processing

Bilevel Knapsack Problem

$$(BKP)\quad \max_{x,y} \; f^1(x,y) = d^1 x + d^2 y \quad \longrightarrow \textit{Leader objective function}$$

$$\text{s.t.}\quad B^1 x + B^2 y \le b^1$$

$$x \in Z_+^{n_1}$$

$$\max_{y} \; f^2(y) = c\,y \quad \longrightarrow \textit{Follower objective function}$$

$$\text{s.t.}\quad A^1 x + A^2 y \le b^2$$

$$y \in \{0,1\}^{n_2}$$

Follower problem

Moore, Bard 1990 => $B^2 = 0$
Dempe, Richter 2000
Brotcorne, Hanafi, Mansi 2011

Relaxation

Definition : Let

(P) $\max\{ f(x) : x \in X\}$

(R) $\max\{ g(x) : x \in Y\}$

Problem R is a relaxation of P if

1) $X \subseteq Y$

2) $\forall\, x \in X : f(x) \leq g(x)$.

Properties

- $v(\mathrm{P}) \leq v(\mathrm{R})$

- Let x^* an optimal solution of R, if x^* is feasible for P and $f(x^*)=g(x^*)$ then x^* is an optimal solution of P

LP-Relaxation

Dantzig (1957)

$$(LP) \quad \begin{array}{ll} \max & cx \\ \text{s.t.} & Ax \leq b \\ & x \in [0,1]^n \end{array}$$

Properties :

- v(MKP) ≤ v(LP)

- Let x^* an optimal solution of LP,

if $x^* \in \{0,1\}^n$ then x^* is an optimal solution of MKP

MIP Relaxation

Hanafi & Wilbaut (06), Glover (06)

- **Let $x^0 \in \{0, 1\}^n$ and $J \subseteq N$**

$$MIP(P,J) \begin{cases} max & cx \\ s.c. & Ax < b \\ & x_j \in \{0,1\} \quad j \in J \\ & x_j \in [0,1] \quad j \in N - J \end{cases}$$

- **Remarks:**
  - ***MIP*(x^0,∅) = LP(P), *MIP*(x^0,N) = *P***
  - ***v*(MI*P*(x^0,*J*)) ≥ *v*(MI*P*(x^0,*J'*) if *J* ⊆ *J'***
- **Stronger bound : *v*(*P*) ≤ *v*(*MIP*(*P*,*J*)) ≤ *v*(*LP*(*P*))**

Semi-Continuous Relaxation

Hanafi & Mansi & Wilbaut (2009)

- Let $J \subseteq N$

$$\textbf{SCR(P, }\alpha\textbf{)} \begin{cases} \max cx \\ \text{s.c. } Ax \leq b \\ x_j \in [0 , \alpha_j] \cup [1-\alpha_j , 1] \;\forall\, j \in N \\ x_j \geq 0 \qquad\qquad j = n+1 \,..\, n' \end{cases}$$

- **Remarks**
  - ✓ $\alpha = e/2 \Rightarrow$ **SCR(P, α) ≡ LP (P)**
  - ✓ $\alpha = (0_J, e_{N-J}/2) \Rightarrow$ **SCR(P, α) ≡ MIP (P,J)**
  - ✓ $\alpha = 0 \Rightarrow$ **SCR(P, α) ≡ P**

Lagrangean Relaxation

Held & Krap (71), Geoffrion (74)

- **Lagrangean Relaxation : $\lambda \geq 0$**
  LR(λ) $\max\{cx + \lambda(b - Ax) : x \in X\}$
- **Lagrangean Dual :**
  (L) $\min\{v(LR(\lambda)) : \lambda \geq 0\}$

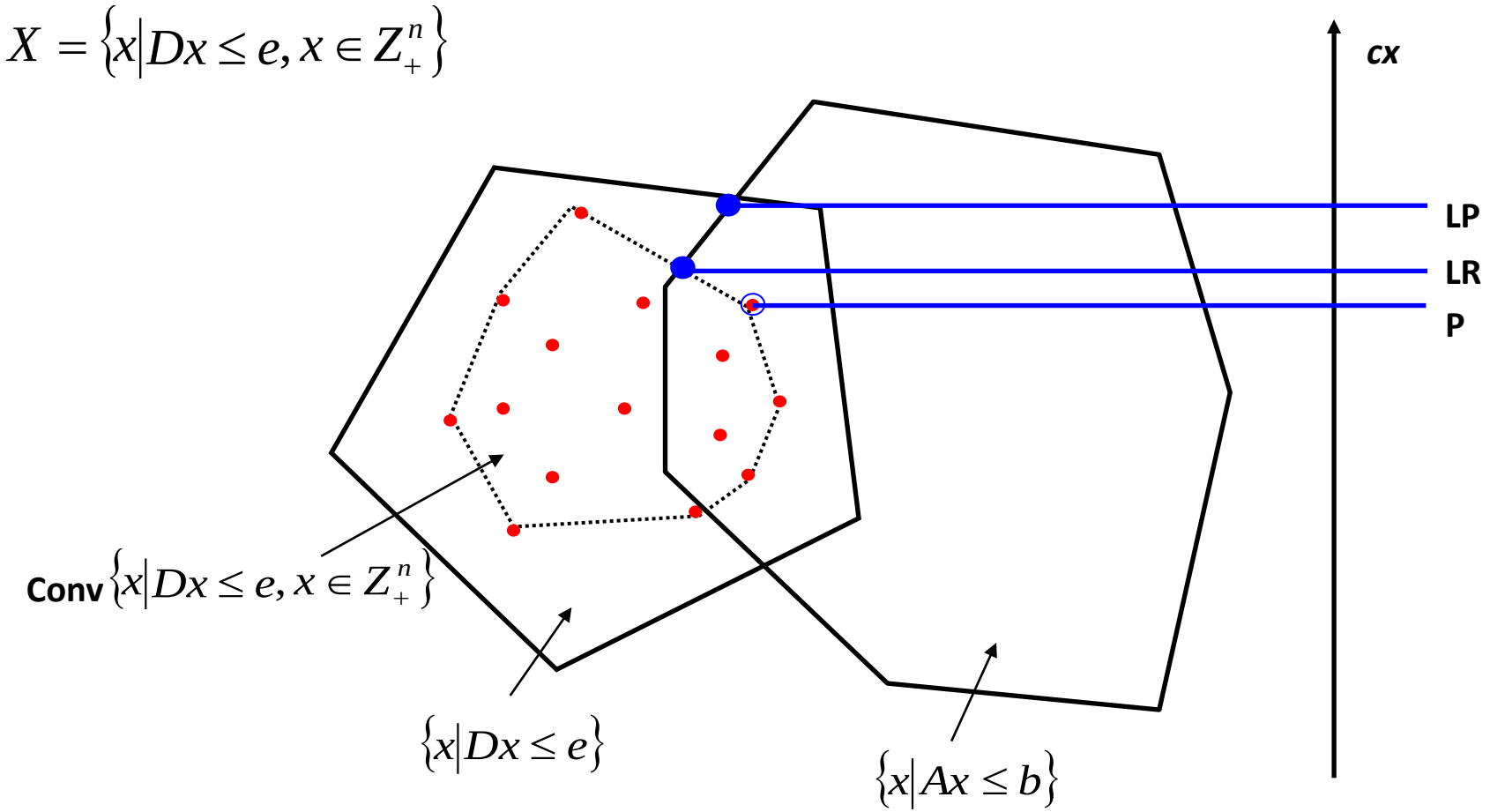
Properties :

- v(MKP) $\leq v$(L) $\leq v$(LR(λ))

- Let x^* be an optimal solution of LR(λ^*), if $\lambda^*(b - Ax^*) = 0$ and x^* is feasible for P then x^* is an optimal solution of P

Geometric Interpretation

$v(\mathrm{L}) = \max\{cx : Ax \leq b, x \in \mathrm{co}(X)\}$



Lagrangean Decomposition

Guignard & Kim (87)

$$Z = \max \quad cx$$
$$Ax \le b$$
$$\boxed{Dx \le e}$$
$$x \in Z_+^n$$

$$Z' = \max \quad cx$$
$$Ax \le b$$
$$\boxed{Dy \le e}$$
$$\boxed{x = y}$$
$$x \in Z_+^n$$
$$y \in Z_+^n$$

$$Z_{LD}(v) = \max \quad cx + v(y - x)$$
$$Ax \le b$$
$$Dy \le e$$
$$x \in Z_+^n$$
$$y \in Z_+^n$$

Lagrangean Decomposition

$$Z_{LD}(v) = \max \quad cx + v(y - x)$$
$$Ax \le b$$
$$Dy \le e$$
$$x \in Z_+^n$$
$$y \in Z_+^n$$

Sub-problem 1

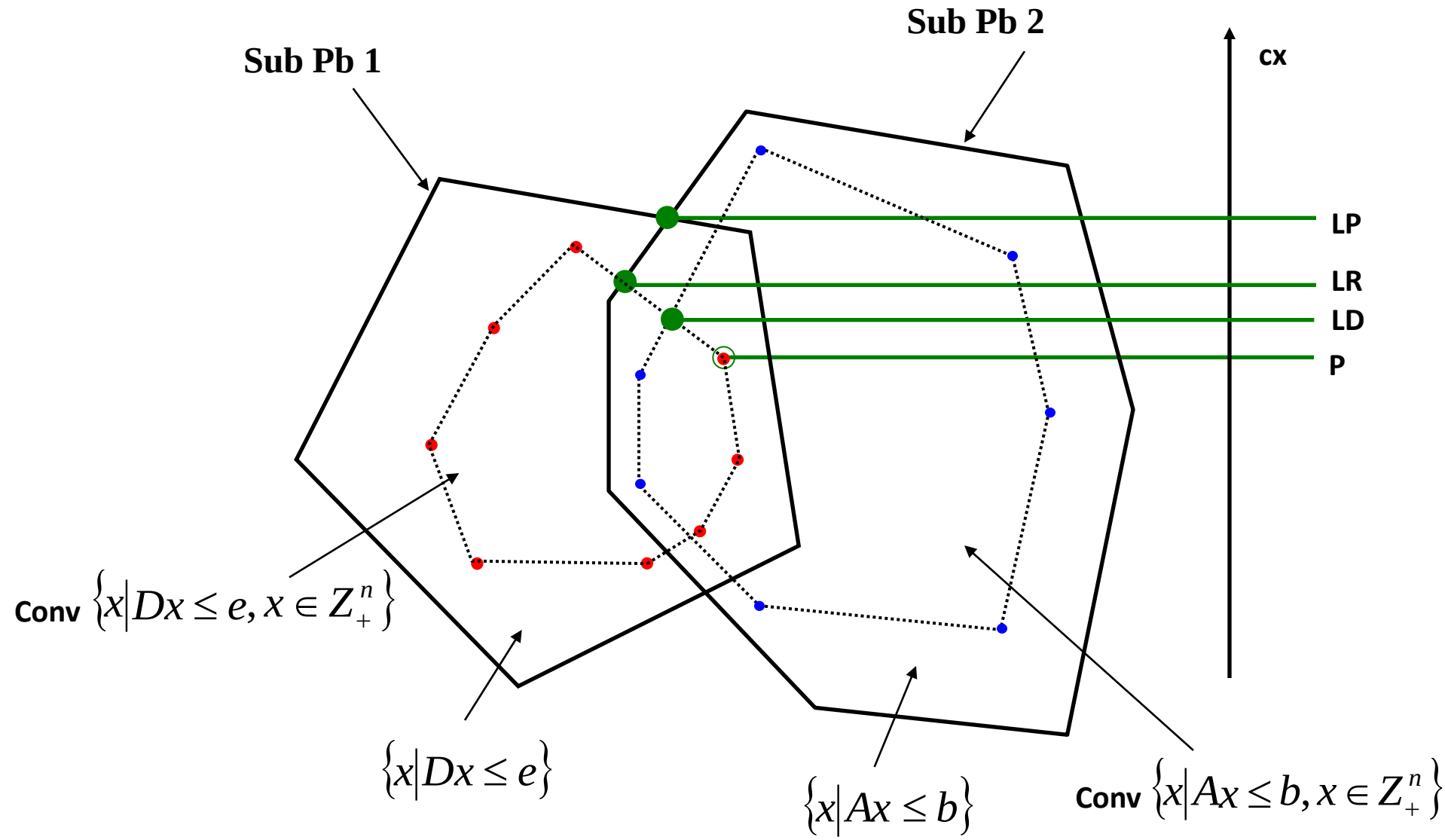
Sub-problem 2

$$Z_{LD1}(v) = \max \quad (c - v)x$$
$$Ax \le b$$
$$x \in Z_+^n$$

$$Z_{LD2}(v) = \max \quad vy$$
$$Dy \le e$$
$$y \in Z_+^n$$

$$Z_{LD} = \min_{v \ge 0} \quad (Z_{LD1}(v) + Z_{LD2}(v))$$

Geometric Interpretation



Theorem

$$v(P) \leq v(LD) \leq v(LR) \leq v(LP)$$

Surrogate Relaxation

- **Relaxation surrogate : $\mu \geq 0$**

  SR(μ) max{cx : μAx $\leq \mu$b, x $\in$ X}

Dual surrogate corresponding :

  (S) min{*v*(SR(μ)) : $\mu \geq 0$}.

Glover (65)

Proprieties :

- *v*(MKP) $\leq$ *v*(S) $\leq$ *v*(SR(μ))

- Let *x be an optimal solution of SR(μ),**

if *x is feasible for MKP then *x** is an optimal solution of MKP**

- (S) min{*v*(SR(μ)) : $\mu \in$ U = {$\mu \geq 0$: $||\mu|| = 1$}}

Composite Relaxation

Greenberg & Pierskalla (70)

- **Composite Relaxation : $\lambda, \mu \geq 0$**

$CR(\lambda, \mu)$ $\quad \max\{cx + \lambda(b - Ax) : \mu Ax \leq \mu b \text{ and } x \in X\}$

Composite Dual :

(C) $\quad \min\{v(CR(\lambda, \mu)) : \lambda, \mu \geq 0\}$

Remarks :

- $v(LR(\lambda)) = v(CR(\lambda, 0))$
- $v(SR(\mu)) = v(CR(0, \mu))$

Comparisons

$$v(P) \leq v(C) \leq v(S) \leq v(L) \leq v(LP)$$

Properties of Relaxation Functions

- **Lagrangean function $v(LR(\lambda))$ is convex and piecewise-linear if X is finite**
- **Surrogate function surrogate $v(SR(\mu))$ is quasi-convex and piecewise-linear if X is finite**
- **Composite function $v(CR(\lambda, \mu))$ is non convex, non quasi-convex**

Relaxation-based Heuristics

- $u = u°$, UB = $g(u)$, LB = $f(x^0)$
- Repeat
  - let $x^*(u)$ an optimal of the current relaxation $R(u)$
  - let $x(u)$ the projection of $x^*(u)$ an the feasible set of P
  - UB = min(UB, $v(R(u))$)
  - LB = max(LB, $f(x(u))$)
  - Update the current multiplier u

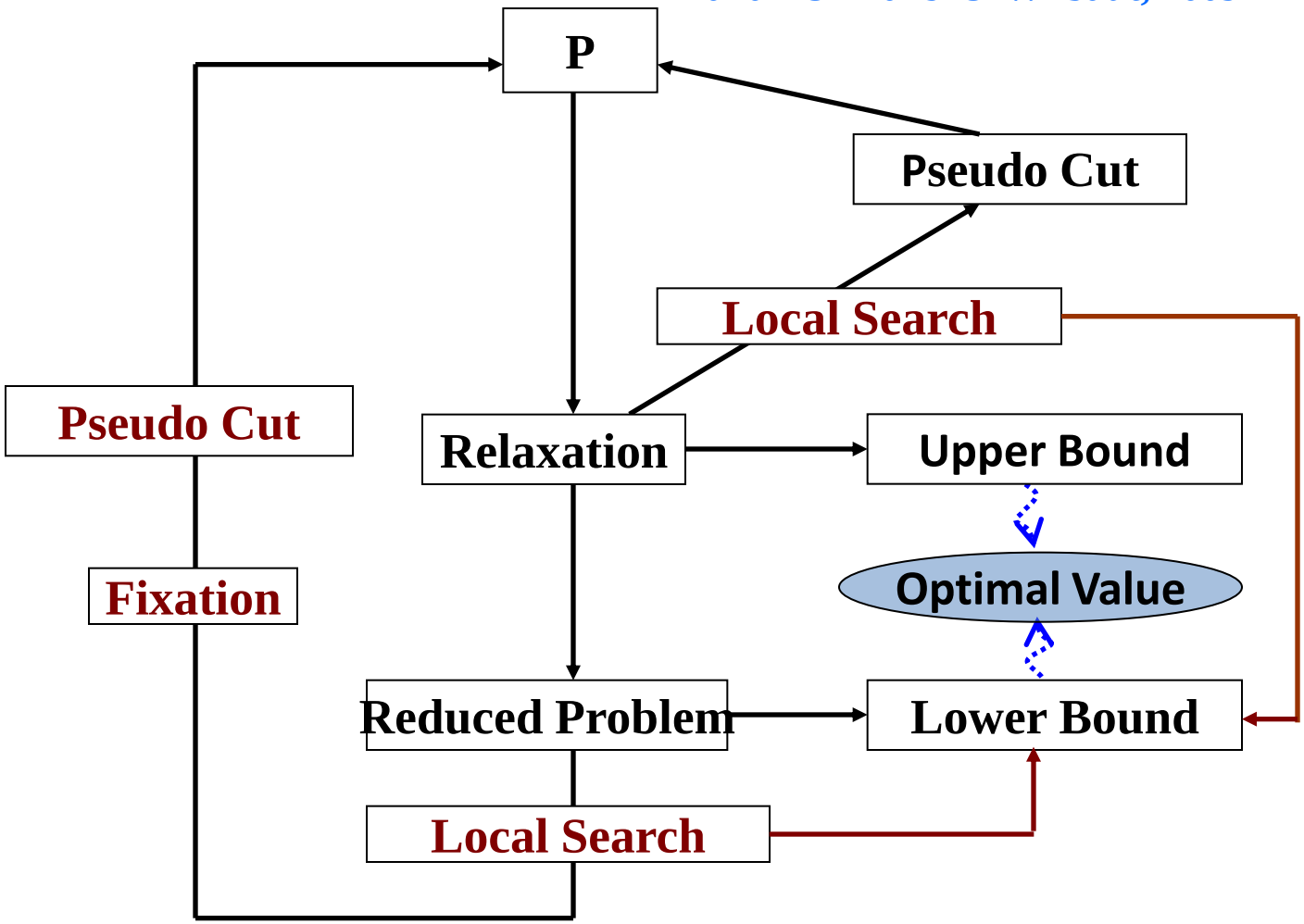
Relaxation-based Heuristics

Hanafi & Wilbaut, 2006

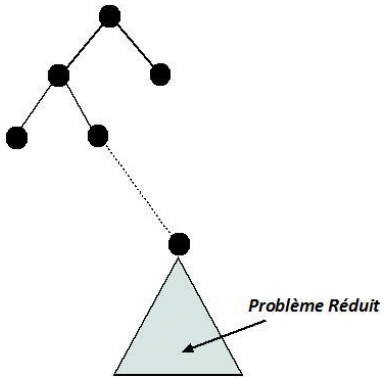
- **Step 1:** Solve one or more **relaxations** of the current problem *P* to generate one or more constraints.
- **Step 2:** Solve one or more **reduced problems** induced by the optimal solution(s) of the previous relaxation(s) to obtain one or more feasible solution(s) of the initial problem.
- **Step 3:** If a stopping criteria is satisfied then return the best lower bound. Else add a generated **constraint**(s) to the problem *P* and return to step 1.

Relaxation-based Heuristics

Hanafi & Mansi & Wilbaut, 2009



Reduced Problem



- Let $x^0 \in \{0, 1\}^n$ and $J \subseteq N$, $x_J = (x_j)_{j \in J}$

$$P(x^0, J) = (P| \; x_J = x^0{}_J)$$

- **Remarks :**
  - $P(x^0, \varnothing) = P$, $v(P(x^0, J)) \geq v(P(x^0, J'))$ if $J \subseteq J'$
  - $x_J = x^0{}_J \Leftrightarrow (e_J - 2\, x^0{}_J) x_J + e_J\, x^0{}_J = 0$
  - $|J| = 1$ If $v(P(e - x^0, \{j\})) \leq cx^0$ then $x^*_j = x^0_j$, x^* an optimal solution

Pseudo-Cut

Let y be a vector in $\{0, 1\}^n$, the following inequality (*) cuts off the solution y without cutting off any other solution in $\{0, 1\}^n$

$$\sum_{j \in J^1(y)} x_j - \sum_{j \in J^0(y)} x_j \leq |J^1(y)| - 1 \qquad (*)$$

where $J^0(x) = \{j \in N : x_j = 0\}$ $J^1(x) = \{j \in N : x_j = 1\}$.

Example:

$$y = (1\ 0\ 1)$$
$$y_1 - y_2 + y_3 \leq 1$$

y_1	0	0	0	0	1	1	1	1
y_2	0	1	0	1	0	1	0	1
y_3	0	0	1	1	0	0	1	1
$y_1 - y_2 + y_3$	0	-1	1	0	1	0	2	1

(*) is called canonical cut in Balas & Jeroslow, 1972

Partial Pseudo Cuts

Let x' be a *partial solution* in $\{0, 1, *\}^n$, the inequality

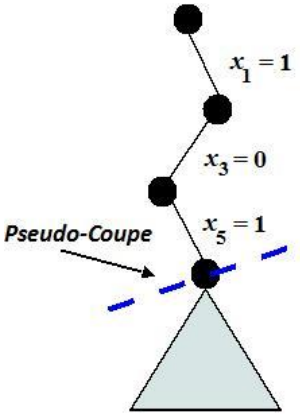
$$d_J(x, x') = (e_J - 2\,x'_J)x_J + e_J\,x'_J \geq 1$$

cuts off x' and all partial solution dominated by x'.

$$J = \{j \in N: x'_j \in \{0,1\}\},\ x_J = (x_j : j \in J)$$

$x' = (1\ *\ 0\ 1\ *)$, $J = \{1, 3, 4\}$

$-x_1 + x_3 - x_4 \geq -1$



Remarks :

- **$d_N(x, x')$ = Hamming distance over $\{0, 1\}^n$**

Iterative LP-based heuristic (ILPH)

Let x^* a feasible solution of P

$v^* = cx^*$; $Q = P$; *stop = False*;

while (*stop = False*) do

Let x^R be an optimal solution of $R(Q)$; J = $J(x^R)$;

Let x^0 be an optimal solution of $(Q / (e_J - 2\, x^R{}_J) x_J \leqq k - e_J\, x^R{}_J)$

if $cx^0 > v^*$ then $x^* = x^0$; $v^* = cx^0$

$Q = (Q \mid (e_J - 2\, x^R{}_J) x_J \geq k + 1 - e_J\, x^R{}_J)$

if $\lfloor cx^R - v^* \rfloor < 1$ then *stop = True*

end while

Theorem : Finite Convergence

The algorithm *ILPH* converges to an optimal solution of the problem in a finite number of iterations ($\leq 3^n$).

Proof :

- $v(P) = max(v(P/dx \leq d_0), v(P/dx \geq d_0 + 1))$
- $d = (e_J - 2\, x^R_J, 0)$ and $d_0 = -\, e_J\, x^R_J$
- **Number of partial solutions = $|\{0, 1, *\}^n| = 3^n$**

Convergence of IPLH

$GK9$ (n=30 m=10)				OR-5.100-4 (n=100 m=5)			
$iter$	$v(LP)$	cx	$\lvert F(x^{LP})\rvert$	$iter$	$v(LP)$	cx	$\lvert F(x^{LP})\rvert$
1	380.3	336	6	1	23724.1	22554	5
2	379.7	368	7	2	23722.7	22983	5
3	379.61	360	8	40	23687.4	23447	18
11	**376.99**	372	13	41	23687.3	**23534**	20
12	376.53	364	7	100	23651.1	23497	24
13	376.42	**376**	9	292	**23534.6**	23497	46

$F(x)$: set of fractional variables of solution x

CPU ≈ 400sec
(Cplex ≈ 10sec)

⇒ Replace the optimal condition by a total number of iterations

Results for MKP : $n = 500$

m	α	Vasquez & Vimont	IRH	$IIRH$
5	0.25	120629.2	120629.2	120630.3
	0.5	219512.7	219512.7	219512.7
	0.75	302363.4	302363.4	302363.4
10	0.25	118628.6	118625.6	11830.8
	0.5	217327.1	217328.3	217330
	0.75	302602.7	302598.9	302604.6
30	0.25	115623.7	115599.5	115605.7
	0.5	216274.7	216255.8	216262.5
	0.75	302446.5	302424.4	302432.5

Average computational time **70 hours** **2 hours**

1 line = average / 10 instances

MKP : Summary (n = 500)

m	#Improvement	#Equality	#Less	Average gap(%)
5	**1**	29	0	0
10	**8**	16	6	0
30	**3**	12	15	0.01

1 **line = results** for 30 instances

Results for MMKP

	C&H			CPLEX		
	#Best	#New Best	Total	#Best	#New Best	Total
HIPL	9	7	16	10	21	31
HIMIP	8	8	16	11	17	28
HIRSC	7	3	10	9	11	20
HIRI	7	5	12	10	19	29

Bi-Objective Max Min Knapsack Problem

n	*VPT*			*ILPH*			
	cpu	*#opt*	*%F*	*cpu*	*#opt*	*%F°*	*%F*
100	32.72	10	07	< 1	10	07	47
200	169.45	9	06	< 1	10	07	50
300	450.18	5	01	10	10	01	37
400	183.18	8	09	8	08	09	63
500	260.45	7	07	15	10	08	61
600	164.00	8	29	11	09	29	56
700	518.09	4	03	19	10	03	56
800	331.27	6	15	30	10	16	66
900	249.09	7	15	18.18	09	16	66
1000	424.54	6	05	166.36	08	06	49

CPU < 900 s

New Hybrid VNDS with IRH

VNDS-IRH: problem P ; initial solution x^*; d ; k_{vnd}

Add objective cut: P = (P | $cx > cx^*$)

While (not Stop1) do

$\bar{x}$ = optimal solution for LP(P)

if ($\bar{x}$ is integer) then $x^* = \bar{x}$, break

Reorder $\delta_j = |x^*_j - \bar{x}_j|$, $j \in B$, so that: $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_{|B|}$

$n_d = |\{j \in B : \delta_j \neq 0\}|$; $k_{step} = [n_d/d]$; $k = |B| - k_{step}$;

While ((not Stop2) and $k \geq 0$) do

J = {1,...,k}; x' = MIPSOLVE(P(x^*,J),x^*(N-J))

if ($cx' > cx^*$) then x^* = VND(P,k_{vnd},x'), break

if ($k - k_{step} > |B| - n_d$) then $k_{step} = \max\{[k/2], 1\}$

$k = k - k_{step}$

Update Stop2

x' = MIPSOLVE(P($\bar{x}$)); $x^* = \text{argmax}\{cx', cx^*\}$

Add pseudo-cut : P = (P | $\delta(J(\bar{x}), x, \bar{x}) \geq 1$)

Update Stop1

Return x^*

Average Results for the OR-Library

Best known values:

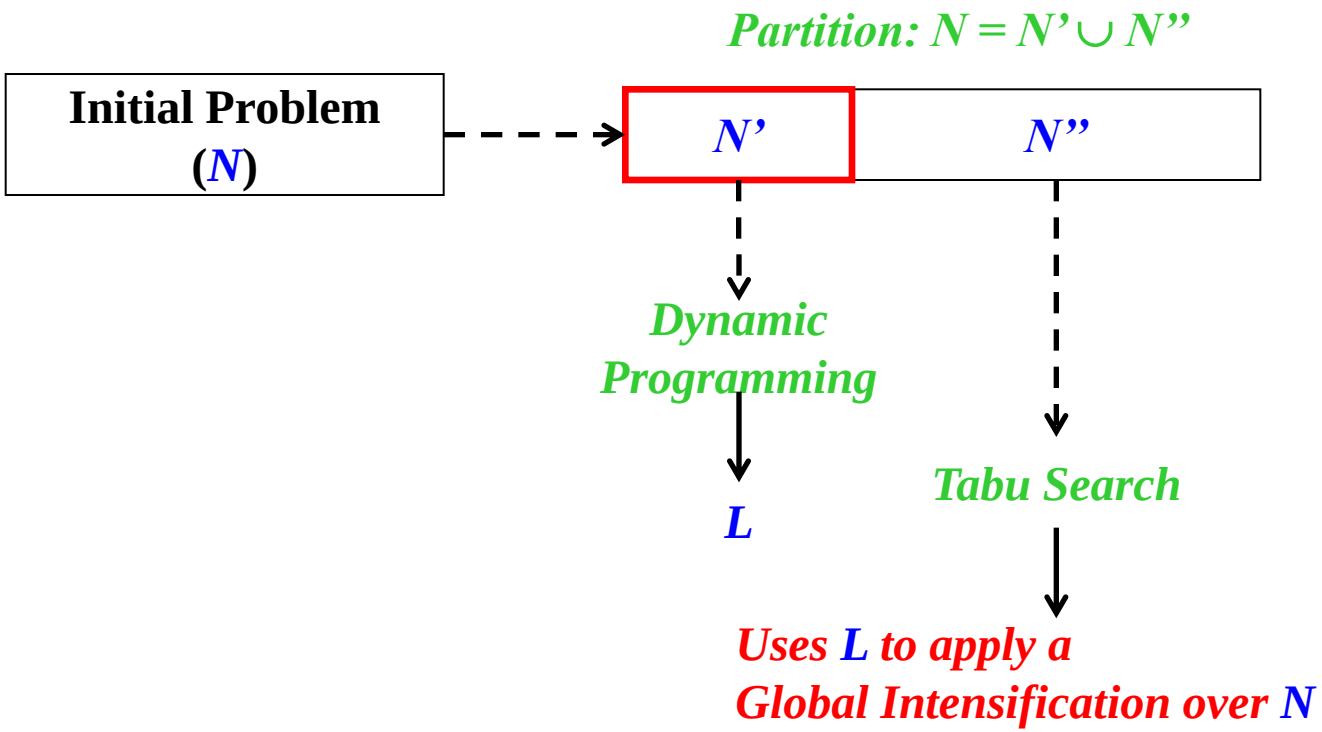
Boussier & Vasquez & Vimont & Hanafi & Michelon (2010)

Hanafi & Wilbaut (2009) Vasquez & Vimont (2005)

Vimont & Boussier & Vasquez (2008) Wilbaut & Hanafi (2009)

		VNDS-IRH	VNDS-IRH2	VNDS-IRH3
10.500	#best	6	5	4
	Avg. Gap	0.009	0.009	0.012
	Avg. CPU*	1703	2057	1508
30.250	#best	21	17	14
	Avg. Gap	0.009	0.013	0.020
	Avg. CPU*	1449	1298	1040
30.500	#best	2	1	3
	Avg. Gap	0.032	0.030	0.031
	Avg. CPU*	2063	1925	2270

Basic Idea of the "Global Intensification" Scheme



➡ Add a second level of intensification in TS

Application to the 0-1 MKP

- **Formulation**

$$(\text{MKP}) \left[\begin{array}{lll} \max & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \forall i \in M = \{1, \ldots, m\} \\ & x_j \in \{0,1\} & j \in N = \{1, \ldots, n\} \end{array} \right.$$

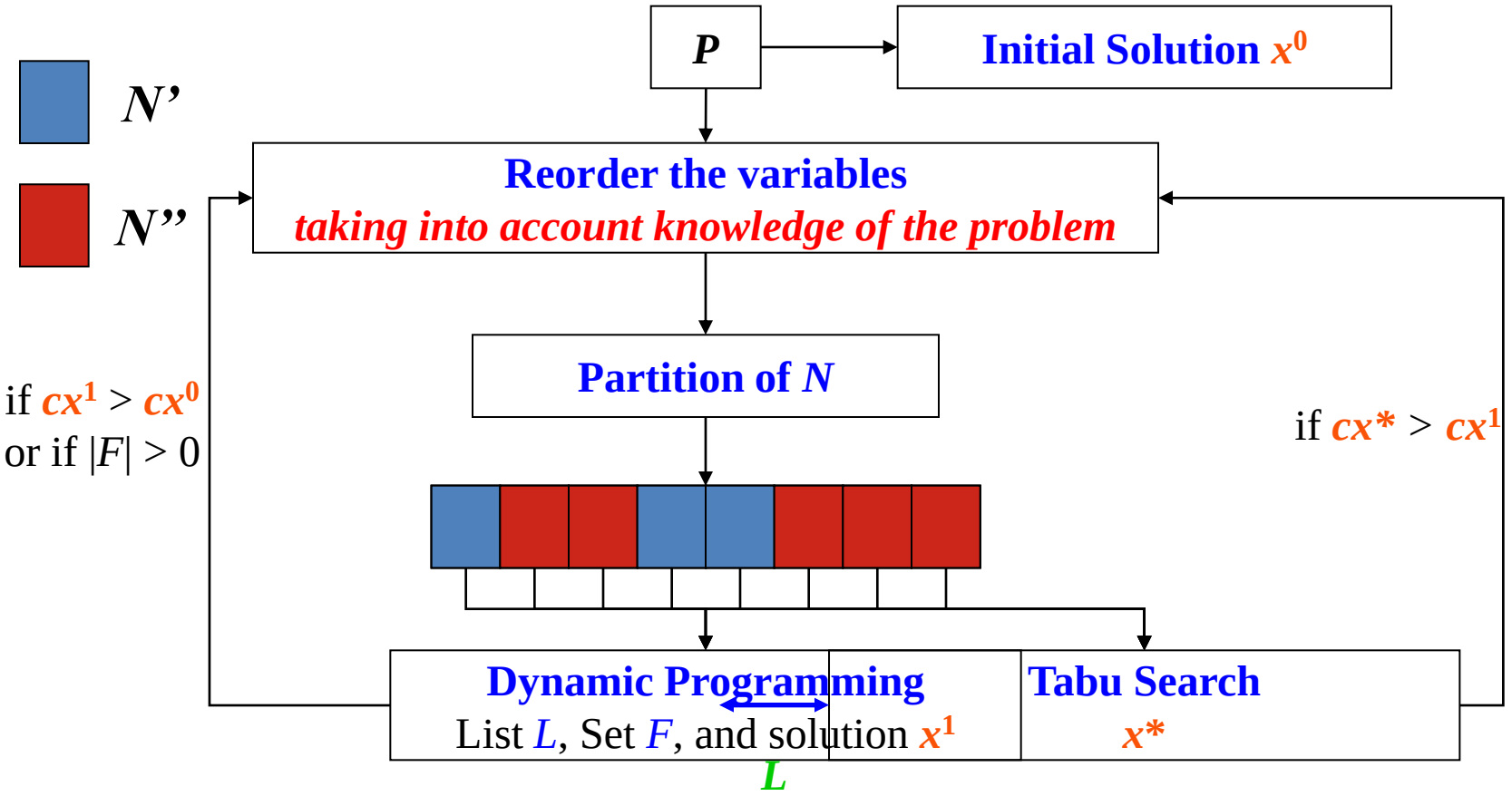
- **Dynamic Programming ($0 \leq k \leq n$ and $0 \leq \beta \leq b$)**

$$(\text{MKP}(k,\beta)) \left[\begin{array}{lll} \max & \sum_{j=1}^{k} c_j x_j & \\ \text{s.t.:} & \sum_{j=1}^{k} a_{ij} x_j \leq \beta_i & i = 1, \ldots, m \\ & x_j \in \{0,1\} & j = 1, \ldots, k \end{array} \right.$$

$$v(\text{MKP}(k,\beta)) = f(k,\beta) = \left\{ \begin{array}{ll} f(k-1,\beta) & \text{if } a_k \not\leq \beta \\ \max\{f(k-1,\beta), c_k + f(k-1,\beta-a_k)\} & \text{if } a_k \leq \beta \end{array} \right.$$

Global Intensification

Let P be an instance of MKP



Global Intensification

- Why (and how) Reordering the variables?

➡ To set some variables at their optimal values

- Let x^0 be an initial solution of P
- Let $u_j \geq v(MKP \,|x_j = 1 - x_j^0)$ for all j in N
- **Lemma**

$$\text{if } v(MKP \,|x_j = 1 - x_j^0) \leq c^T x^0 \text{ then } x_j^* = x_j^0$$

- Reorder the variables such that:

$$u_1 \geq u_2 \geq \ldots \geq u_n$$

Global Intensification

- Property 1 (**Optimality**):
  Let $l_j = v(MKP \,|\, x_k = x_k^0, \forall k = j+1, \ldots, n)$
  if $\exists j \in \{1, 2, \ldots, n\}$ such that $l_j \geq u_{j+1}$
  then $v(MKP) = l_j$

- Proof: let x^* be an optimal solution and $j \in N$ such that $l_j \geq u_{j+1}$

  First case: if $x_k^* = x_k^0, \forall k = j+1, \ldots, n$ => OK

  Second case: if $\exists k \in \{j+1, \ldots, n\}, x_k^* \neq x_k^0$

$$u_{j+1} \geq u_k \text{ since } u_1 \geq \ldots \geq u_n$$

$$u_k \geq v(MKP \,|\, x_k = 1 - x_k^0) = v(MKP) \text{ since } x_k^* \neq x_k^0$$

$$l_j \geq u_{j+1} \geq v(MKP)$$

Global Intensification

- Remark: $l_1 \leq l_2 \leq \ldots \leq l_n$ (1)
- Property2 (**Reduction**):

  If $\exists h, k$ with $h < k < n$ and $l_h \geq u_k$
  then $x_j^* = x_j^0, \forall j \in \{k, \ldots, n\}$

- Proof: suppose $\exists\ h < k < n$ with $l_h > u_k$ (H)

  then: $h < k$ $\Rightarrow h \leq k - 1$

  $\Rightarrow l_h \leq l_{k\text{-}1}$ from (1)

  $\Rightarrow u_k \leq l_h \leq l_{k\text{-}1}$ from (H)

  $\Rightarrow v(\text{MKP}) = l_{k\text{-}1}$ (Prop.1)

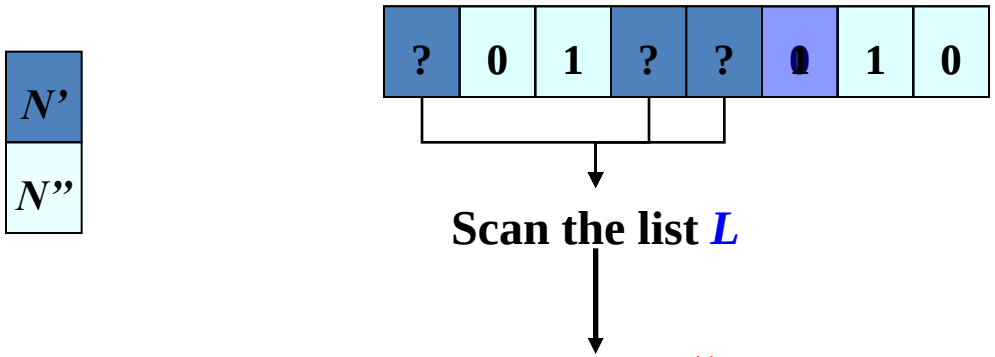
Global Intensification

- Using the List L in Tabu Search
  - **Neighborhood** of the current solution *x''* on *N''*

➡ Drop / Add one item

$$V(x'') = \{y \in \{0,1\}^{|N''|} \,\Big|\, Ay \leq b \text{ and } \sum_{j \in N''} \left|x''_j - y_j\right| \leq 1\}$$

  - **Evaluation of a move**



$$max\{v : \beta \geq \sum_{j \in N''} a_j x''_j, (v,\beta) \in L\} + \sum_{j \in N''} c_j x''_j$$

Global Intensification

- Experiments over OR-Library instances
  - The solution quality **increases during the search** (initial solution, dynamic programming, tabu search)
  - The Global Intensification process obtains **better** results than a "*simple*" **Tabu Search**
  - The results were **better** than those obtained by **CPLEX**
  - Reduction rules were **efficient when $m << n$**

Global Intensification: Conclusions

- Introduction of a **second level of intensification** in Tabu Search
- The Global Intensification **can be applied** for a class of combinatorial optimization problems, whose the structure is adapted to Dynamic Programming treatment
- **Efficient cooperation** between Dynamic Programming and Tabu Search (for the MKP)

Decomposition

Glover 1965, Vasquez, 2000, Hanafi et al. 2011

Let x^* be best known solution

$$X = \{x : Ax \leq b, cx \geq cx^*+1\}$$

$$k_{min} = \min\{ex : x \in X \cap [0,1]^n\}$$

$$k_{max} = \max\{ex : x \in X \cap [0,1]^n\}$$

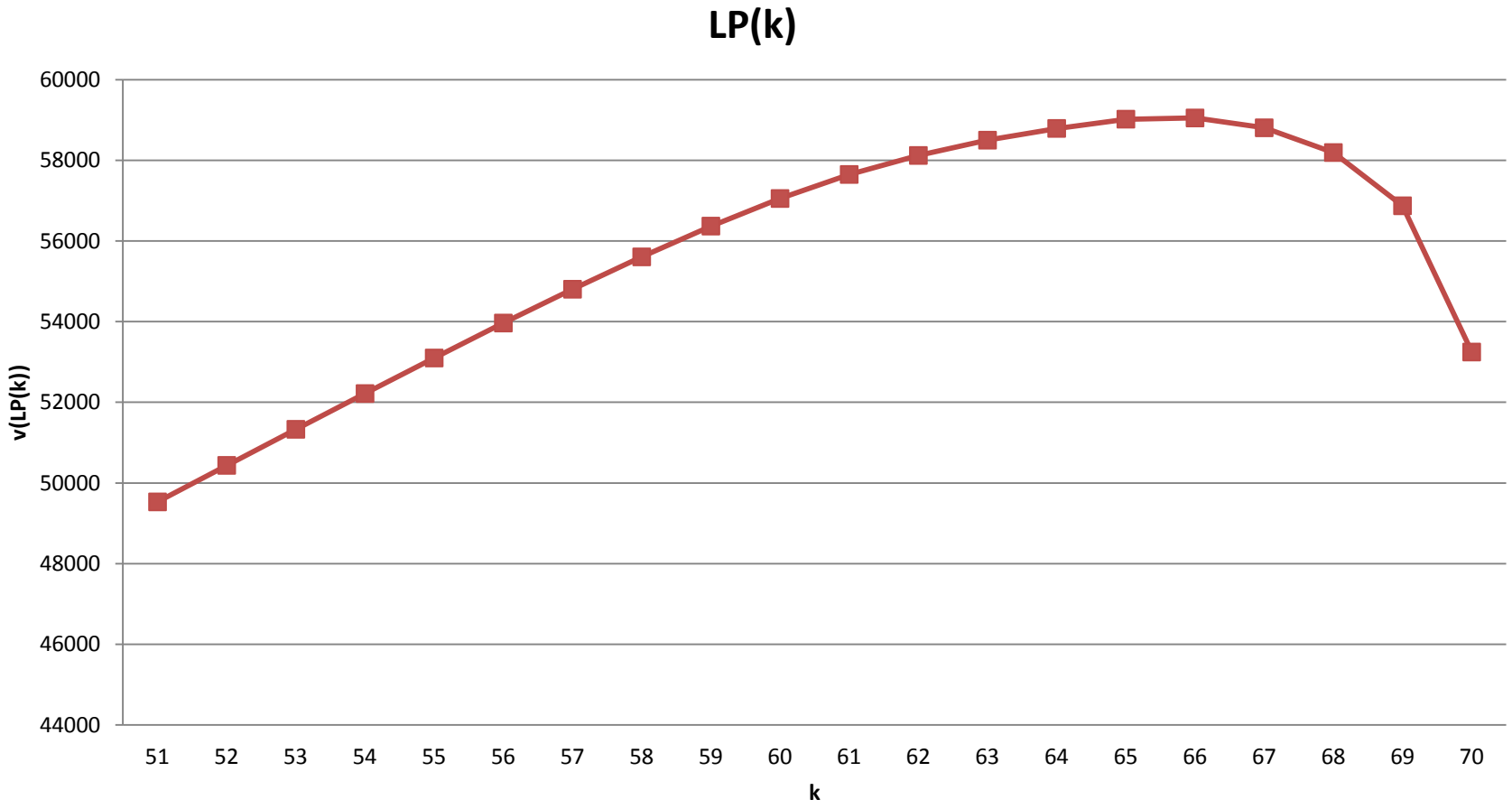
$$\text{MKP}(k)\ \max\{cx : ex = k\ ,\ x \in X \cap \{0,1\}^n\}$$

We have:

$$v(\text{MKP}) = \max\{v(\text{MKP}(k)) : k = \lceil k_{min} \rceil, \ldots, \lfloor k_{max} \rfloor\}$$

$$\text{LP}(k) = \text{LP}(\text{MKP}(k))$$

$v(\mathrm{LP}(k))$ is concave



Tabu Search with Relaxation for MKP

Let $k = \lfloor ey \rfloor$ and $h = \lceil ey \rceil$ where y = R(MKP); $x^* = 0$;

do{

Stop = False; y = Relaxation(MKP(k));

if $cy > cx^*$ then

x = Tabu-Search(MKP(k), y); $k = k - 1$;

$x^* = argmax\{cx, cx^*\}$; Stop = True;

y = Relaxation(MKP(h));

if $cy > cx^*$ then

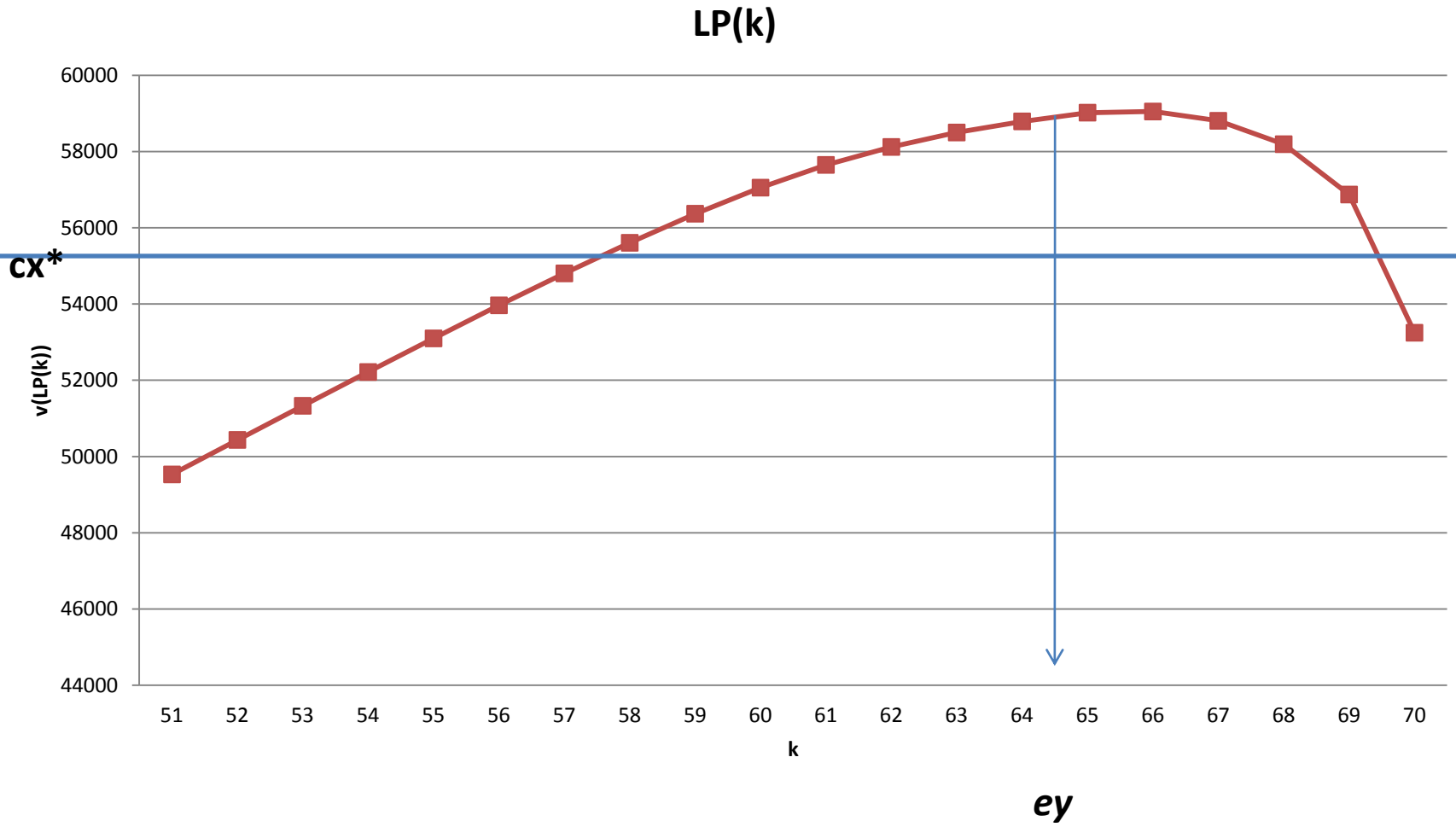
x = Tabu-Search(MKP(h), y); $h = h + 1$;

$x^* = argmax\{cx, cx^*\}$; Stop = True;

}while Stop;

Relaxation = {LP or MIP}

Hyperplanes exploration order



Tabu Search for MKP(k)

Neighborhood Vasquez & Hao, 2001; Vasquez & Vimont, 2005;

- $y = \mathrm{R}(k)$; $\sigma = 2(ey - k)$
- $N(x) = \{x' \in \{0,1\} : \|x - x'\|_1 = 2$
  $\|x - y\|_1 \leq \sigma\ ,\ ex' = k\}$

Dynamic Tabu List Glover, 1990

Reverse Elimitation Method Dammeyer & Voß, 1993; Glover & Laguna, 1997; Hanafi & Fréville, 1999;

Aspiration

– New best solution even $\|x' - y\|_1 > \sigma$

Results CB30.500-0

```
CB30.500_0 k=130 hwf=0 hwb=0 ub=116601.4

.     2 115332      0 1224 1653     0 sec.

.     2 115604 115332 1127 1249     0 sec.

*    40 115637 115637 1326 1351     0 sec.

*     2 115653 115653 1492 1517     0 sec.

*     4 115688 115688 1656 1681     0 sec.

*     6 115695 115695 1740 1765     0 sec.

*    26 115708 115708 1781 1806     1 sec.

.   468 115713 115708 1722 1806     3 sec.

.  1014 115809 115713 1641 1747     4 sec.

.     2 115832 115809 1594 1666     5 sec.

*  9906 115862 115862 1752 1777    28 sec.

.   296 115869 115862 1750 1777    38 sec.

*   108 116056 116056 1821 1846    54 sec.
```

Results CB30.500

	lb	k
cb30.500_0	116056	130
cb30.500_1	114810	128
cb30.500_2	116721	128
cb30.500_3	115329	127
cb30.500_4	**116565**	**128**
cb30.500_5	115741	131
cb30.500_6	114181	128
cb30.500_7	114348	128
cb30.500_8	115419	128
cb30.500_9	117116	128
cb30.500_10	218104	251
cb30.500_11	214648	251
cb30.500_12	215978	250
cb30.500_13	217910	251
cb30.500_14	215689	251

	lb	k
cb30.500_15	215890	253
cb30.500_16	215907	253
cb30.500_17	216542	254
cb30.500_18	217340	253
cb30.500_19	214739	252
cb30.500_20	301675	375
cb30.500_21	300055	374
cb30.500_22	305087	375
cb30.500_23	302032	375
cb30.500_24	304462	375
cb30.500_25	297012	374
cb30.500_26	303364	373
cb30.500_27	307007	377
cb30.500_28	303199	375
cb30.500_29	300572	376

Some Related Works

- Canonical Cuts on the Unit Hypercube, Balas & Jeroslow, 1972
- 0-1 IP with many variables and few constraints, Soyster, Lev, Slivka, 1978
- A hybrid approach to discrete mathematical programming, Marsten, Morin, 1978
- A new hybrid method combining exact solution and local search, Mautor, Michelon, 1997
- Parametric Branch and Bound, Glover, 1978
- Large neighborhood search, Shaw, 1998
- Local branching, Fischetti, Lodi, 2002

Some Related Works

- Relaxation induced neighbourhood search, Danna, Rothberg, Pape, 2003
- Adaptive memory projection method, Glover, 2005
- Feasibility Pump, Fischetti, Glover, Lodi, 2005
- Relaxation guided VNS, Puchinger, Raidl, 2005
- Global intensification using dynamic programming, Wilbaut, Hanafi, Fréville, Balev, 2006
- Convergent Heuristics for 0-1 MIP, Wilbaut, Hanafi, 2006
- Variable Neighbourhood Decomposition Search, Lazic, Hanafi, Mladenovic, Urosevic, 2009
- Inequalities and Target Objectives in Metaheuristics for MIP, Glover, Hanafi, 2009