A branch-and-bound algorithm to solve large scale integer quadratic multi-knapsack problems

Dominique Quadri**, Eric Soutif*, Pierre Tolla**

**Lamsade, University Paris Dauphine

*Cedric, CNAM Paris

SOFSEM 2007, January 20-26 2007, Harrachov, Czech Republic

Talk overview

- The problem
- Existing branch-and-bound
- The proposed branch-and-bound
- Computational results
- Conclusions and future works

The problem

$$
(QMKP)\ \left\{\begin{array}{ll}
\max & f(x) = \sum_{i=1}^{n}(c_i x_i - d_i x_i^2) = \sum_{i=1}^{n} f_i(x_i) \\
s.t. & \left| \begin{array}{lll}
\sum_{i=1}^{n} a_{ji} x_i \leq & b_j & j = (1, 2, ..., m) \\
0 \leq x_i \leq u_i & integer & \\
where & & \\
c_i \geq 0,\ d_i \geq 0, a_{ji} \geq 0, b_j \geq 0, u_i \leq (c_i/2d_i) & &
\end{array}\right.
\end{array}\right.
$$

- We are interested in an **integer quadratic multi-knapsack problem** with a **separable** objective function.
- NP-hard problem
- Our aim : develop a practical method to solve to optimality $(QMKP)$

Notations

- Let (P) be a pure integer or $0 - 1$ program
- Let $(\overline{P})$ be the LP relaxation of (P)
- $Z[P]$: optimal value of the problem (P)
- $Z[\overline{P}]$: optimal value of $(\overline{P})$.

Branch-and-bound algorithm

- A feasible solution
- A tight upper bound at each node of the search tree
- Before starting the branch-and-bound procedure : preprocessing techniques

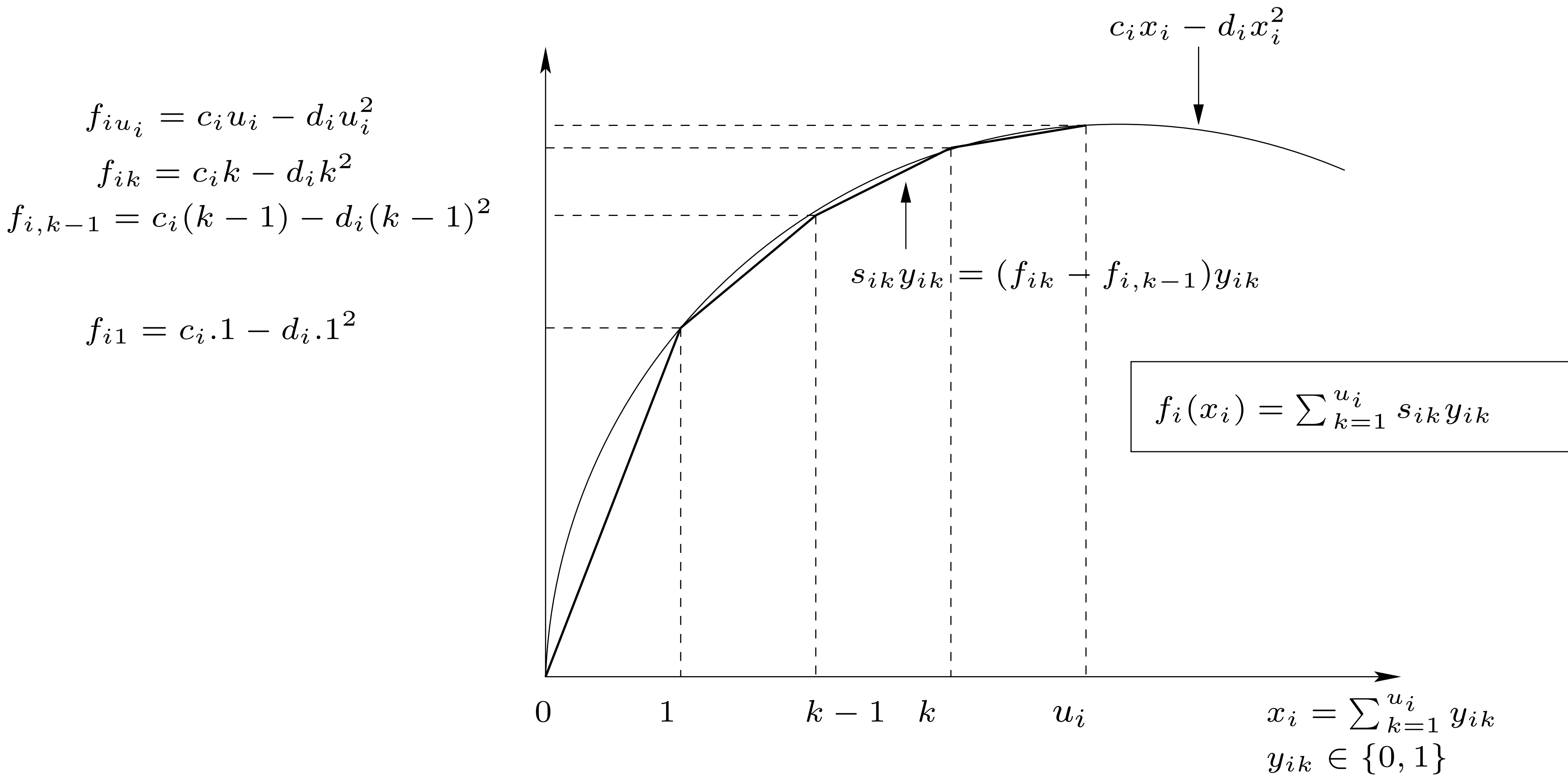
Standard $B\&B$ approach (SBB)

- Quadratic concave objective function subject to m linear constraints
- $Z[\overline{QMKP}]$: upper bound
- *Cplex9.0*.

A 0-1 linearization $B\&B$ (LBB)

- Transform $(QMKP)$ into a 0-1 equivalent problem :
  - direct expansion : re-write the integer variables into 0-1 variables
  - piecewise linear interpolation
- Mathur and Salkin (1983) : branch-and-bound to solve the single constraint integer quadratic knapsack (QKP)

A 0-1 linearization $B\&B$ (LBB)



A 0-1 linearization $B\&B$ (LBB)

$$(MKP)\left\{\begin{array}{ll} \max & \sum_{i=1}^{n}(\sum_{k=1}^{u_i} s_{ik}y_{ik}) \\ s.t. & \left| \begin{array}{l} \sum_{i=1}^{n}(a_{ji}\sum_{k=1}^{u_i} y_{ik}) \leq b_j \\ (j=1,2...,m) \\ y_{ik}\in\{0;1\} \end{array}\right. \end{array}\right.$$

where

- $x_i = \sum_{k=1}^{u_i} y_{ik}$, $y_{ik} \in \{0;1\}$,
- $s_{ik} = f_{ik} - f_{i,k-1}$,
- $f_{ik} = c_i k - d_i k^2$.

<u>Proposition</u> : $Z[\overline{MKP}] \leq Z[\overline{QMKP}]$

Djerdjour et al. algorithm UB

$(QMKP)$ and $Z[QMKP]$ the optimal value

Equivalence

(MKP) equivalent 0-1 formulation of $(QMKP)$

$Z[MKP]$ the optimal value

$Z[MKP] = Z[QMKP]$

Surrogate relaxation $\quad w \geq 0$

$Z[\overline{MKP}] \qquad \leq \qquad \min_w Z[\overline{KP, w}]$

$(\overline{KP, w})$ surrogate relaxation of $(\overline{MKP})$

$Z[\overline{KP, w^*}]$ the optimal value of the continuous relaxation of (KP, w^*)

Djerdjour et al. algorithm (DMS)

- Surrogate relaxation : transform the m constraints of $(\overline{MKP})$ into one constraint (called surrogate constraint) ;
- Surrogate multiplier : $w = (w_1, ..., w_j, ..., w_m) \geq 0$;
- $(\overline{MKP})$ becomes :

$$(\overline{KP,w}) \begin{cases} \max & \sum_{i=1}^{n}(\sum_{k=1}^{u_i} s_{ik}y_{ik}) \\ s.t. & \left| \begin{array}{l} \sum_{i=1}^{n}[\sum_{j=1}^{m} w_j a_{ji}] \sum_{k=1}^{u_i} y_{ik} \leq \sum_{j=1}^{m} w_j b_j \\ y_{ik} \in [0;1] \end{array} \right. \end{cases}$$

- $Z[\overline{MKP}] \leq Z[\overline{KP,w}]$
- How to find a good surrogate multiplier w^* ?

How to find w^* ? (DMS)

- Let us consider : $Z[\overline{KP, w}]$
- Solving $(SD) \ = \min_{w \geq 0} Z[\overline{KP, w}]$
- (SD) is called the surrogate dual
- Problem easy to solve :
  - The objective function of (SD) is quasi-convexe
  - Local descent method
  - w^* is a global mimimum

The proposed $B\&B$

- Improving the upper bound of (DMS)
  - Decreasing the computational time
  - Getting a tighter upper bound
- A heuristic to compute a feasible solution
- Pre-processing procedures

Decreasing the computational time

- Proposition 1
  If w^* is the dual optimal solution of $(\overline{MKP})$ then the optimal value of $(\overline{MKP})$ is equal to the optimal value of $(\overline{KP, w^*})$ that is :
  $Z[\overline{MKP}] = Z[\overline{KP, w^*}]$

  - Decreasing the computational time of w^*

  - w^* : dual optimal solution of $(\overline{MKP})$

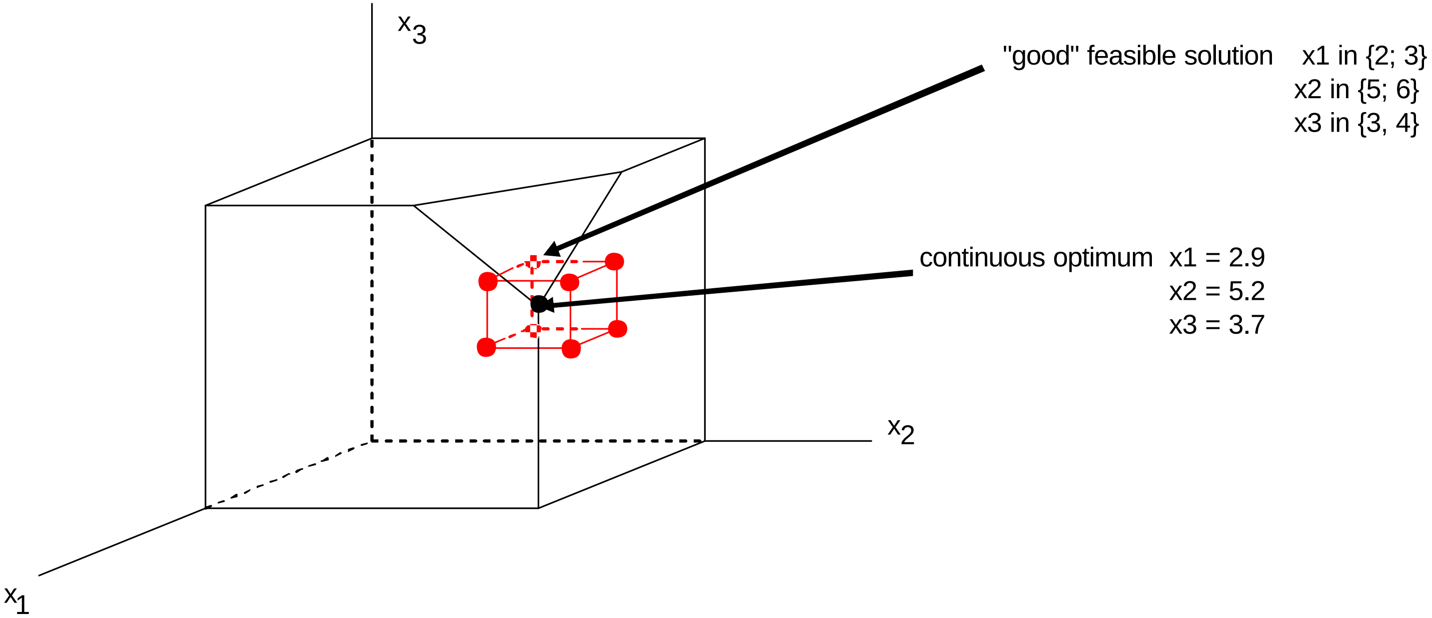
Getting a tighter upper bound

- Improving the upper bound value
  - $Z[KP, w^*]$: an improved upper bound
  - Analytically the upper bound is improved.

Analytical comparison of the upper bounds

$+\infty$

Problem	Upper bound
$(\overline{QMKP})$	$Z[\overline{QMKP}]$ [LP relaxation]
$(\overline{KP, w^*})$ $(\overline{MKP})$	[Djerdjour at al. 1988] $Z[\overline{KP, w^*}]$ $=$ $Z[\overline{MKP}]$ [Linearized formulation]
(KP, w^*)	$Z[KP, w^*]$ [Our approach]
(MKP) $\Updownarrow$ $(QMKP)$	Linearized optimum $=$ Quadratic optimum


x3
"good" feasible solution x1 in {2; 3}
x2 in {5; 6}
x3 in {3, 4}
continuous optimum x1 = 2.9
x2 = 5.2
x3 = 3.7
x2
x1

Pre-processing procedures

- Detecting some redundant constraints
- Reducing the bounds of integer variables : contraints pairing procedure, Hammer et al. (1975).
- Simultaneously fixing some 0-1 variables to 0

Computational results

- square problems ($n = m$)
- problems are randomly generated in the interval $[0, 100]$ according to an uniform law
- average $\%$ of pure integer variables : 40% for squared problems
- average value of u_i : 22 for squared problems.

Average CPU time of the 4 $B\&B$

n	m	Our BB	LBB	SBB	DMS
100	100	1.5	1.3	7.8	208.257
500	500	29.3	120.1	19.1	-
1000	1000	50.5	264.4	282.3	-
1500	1500	183.7	392.5	1178.4	-
2000	2000	305.2	1369.4	2557.9	-

"-" : optimum not reached in a limit time of 3 hours

Analyzing the computational results

The improvement capability of our $B\&B$ can be explained by three features, namely :

1. the feasible solution

2. the upper bound

3. the pre-processing procedures

The upper bound

		Av. deviation to the opt. (%)			CPU time (sec.)			
		Our BB	LBB=DMS	SBB	Our BB	LBB	SBB	DMS
n	m							
100	100	8.2	9.5	16.9	0.0	0.0	0.0	0.3
500	500	7.5	7.9	12.9	0.2	0.1	7.3	9.0
1000	1000	21.7	23.0	32.2	0.5	0.5	58.2	37.9
1500	1500	23.9	24.6	37.8	1.6	1.5	184.5	86.6
2000	2000	36.2	36.9	53.0	3.6	3.4	421.3	157.8

The pre-processing procedures

- Detecting some redundant constraints : on average 52% of the constraints may be removed
- Reducing the bounds of integer variables : the average proportion of pure integer variables has decreased from 40% to 21.02%
- Simultaneously fixing some 0-1 variables to 0 : 50.25% of 0-1 variables are fixed

Conclusions and future works

- Conclusions
  - Our $B\&B$ allowed us to solve large scale instances : up to 2000 variables within 306 s on average (largest problems)
  - (LBB) is a possible alternative to solve $(QMKP)$
  - (SBB) and (DMS) can be used only for small instances
- Future works
  - Improve our upper bound
  - Solve a nonseparable quadratic multi-knapsack problem