# A parametric programming approach to redefine the global configuration of resource constraints of 0-1-Integer Linear Programming problems.

Alejandro Crema
email: alejandro.crema@ciens.ucv.ve
Paseo Los Ilustres, Facultad de Ciencias,
Universidad Central de Venezuela, Caracas, Venezuela

October 2016

**Abstract**

A mathematical programming approach to deal with the global configuration of resource constraints is presented. A specialized parametric programming algorithm to obtain the pareto set for the biobjective problem that appears to deal with the global configuration for 0-1-Integer Linear Programing problems is presented and implemented. Computational results for Multiconstrained Knapsack problems and Bounded Knapsack problems are presented.

**Keywords** Integer programming, Multiple objective programming, Parametric programming, Bottlenecks, Resource Constraints.

# 1 Introduction.

A new definition of bottleneck has been presented recently as follows ([1],[2]): "A bottleneck is a modifiable specification of resources that by changing its value, the best achievable performance of the system can be improved". This new definition is more general than some previous based on average shadow prices ([3],[9],[10],[12],[13]). The authors consider the case in which the decision maker may redefine the system global configuration and then define a problem with two objectives: maximization of the original objective and minimization of the price of the modification. In order to consider nonlinear systems an evolutionary algorithm is proposed to obtain an approximation of the pareto set.

In section 2 we present a mathematical programming approach to deal with the global configuration in a 0-1-Mixed Integer Linear Programming (0-1-MILP) problem. In section 3 we present a specialized parametric programming algorithm to obtain the pareto set for the biobjective problem that appears to deal with the global configuration of resource constraints for 0-1-Integer Linear Programing (0-1-ILP) problems. Computational results are presented in section 4. Finally the conclusions and some extensions are presented is section 5.

A few words about our notation: If $S$ is an optimization problem then $v(S)$ is its optimal value (if it exists) and $F(S)$ is its set of feasible solutions. If we write $S(\theta, \cdots, \gamma)$ is a problem in $(x, \cdots, y)$ that means that $x, \cdots, y$ are the variable vectors and $\theta, \cdots, \gamma$ are data vectors that may change from one problem to another. The rest of the data for $S$ are fixed and that must be clear in the context. A vector or matrix with zeros will be denoted 0. If a property is valid for $k = 1, \cdots, K$ we may write that the property is valid $\forall k$ when $K$ is known in the context. If $D$ is a matrix with $D \in \Re^{l \times r}$ its rows will be denoted $D_1, \cdots, D_l$ and its columns will be denoted $D^1, \cdots, D^r$. A uniform distribution in $(a, b)$ will be denoted $U(a, b)$.

# 2 A mathematical programming approach to redefine the global configuration

Let $P$ be a 0-1-MILP problem in $(x)$ defined as:

$$(P) \;\; max \;\; c^t x \;\; s.t.$$

$$\hat{A}x \leq \hat{b}, \;\; Ax \leq b, \;\; x \geq 0$$

$$x \in \Re^n, \;\; x_j \in \{0, 1\} \;\; \forall j \in J$$

where $c \in \Re^n, b \in \Re^m, A \in \Re^{m \times n}, \hat{A} \in \Re^{s \times n}, \hat{b} \in \Re^s$ and $J \subseteq \{1, \cdots, n\}$.

Let $\Omega = \{x : \hat{A}x \leq \hat{b},\ x \in \Re^n,\ x \geq 0,\ \ x_j \in \{0,1\}\ \ \forall j \in J\}$.

Let us suppose that $F(P) \neq \emptyset$ and $\Omega$ is a compact set.

Let us suppose that $Ax \leq b$ are resource constraints that may be redefined. Some examples are: (i) the availability of some resources may be increased (the capacity of a knapsack, the capacity of a plant, the number of available machines, the available time to use a machine, etc...) and (ii) some coefficients of the matrix in the resources constraints may be reduced (the time that we need to process a job in a machine, the units of a resource to do a job, etc...).

Let $r \geq 1$. Let $\underline{\theta} \in \Re^r$ and $\overline{\theta} \in \Re^r$ with $\underline{\theta} \leq 0 \leq \overline{\theta}$. Let $\theta \in [\underline{\theta}, \overline{\theta}]$.

In this paper we consider linear changes as follows:

Let $q^j \in \Re^r$ $(j = 1, \cdots, n), d^i \in \Re^r$ $(i = 1, \cdots, m)$ and $Q^{ij} \in \Re^r$ $(i = 1, \cdots, m)(j = 1, \cdots, n)$ and let:

$$c(\theta)_j = c_j + {q^j}^t\theta \ \ \forall j$$

$$b(\theta)_i = b_i + {d^i}^t\theta \ \ \forall i$$

$$A(\theta)_{ij} = A_{ij} + {Q^{ij}}^t\theta \ \ \forall (i,j)$$

The mathematical programming problem in $(x)$ with the configuration redefined by $\theta \in \Re^r$ is defined as follows:

$$(P(\theta)) \ \ max \ \ c(\theta)^t x \ s.t.$$

$$A(\theta)x \leq b(\theta)$$

$$x \in \Omega$$

which is:

$$(P(\theta)) \ \ max \ \ c^t x + \sum_{j=1}^{n} {q^j}^t \theta x_j \ \ s.t.$$

$$A_i x + \sum_{j=1}^{n} {Q^{ij}}^t \theta x_j \leq b_i + {d^i}^t\theta \ \ \forall i$$

$$x \in \Omega$$

The paper may be rewritten without any problem if we add additional constraints to limit the valid changes.

Let $p \in \Re^r$. Let us suppose that the price of the configuration redefined by $\theta$ is $p^t\theta$.

The biobjective mathematical programming problem in $(x, \theta)$ suggested in [1] and [2] is defined in this case as follows:

$$(BI)$$

$$max\ c(\theta)^t x,\ \ min\ \ p^t\theta\ s.t.$$

$$x \in F(P(\theta)),\ \ \underline{\theta} \leq \theta \leq \overline{\theta}$$

Since $F(P(0)) \neq \emptyset$ it follows that $F(BI) \neq \emptyset$.

We follow a standard linearization procedure ([7]) to deal with the terms $x_j\theta\ \forall j \in J$ in order to rewrite $BI$ as a 0-1-Mixed Integer Bilinear Programming (0-1-MIBLP) problem in $(x, \theta, \delta)$ with the bilinear terms restricted to the continuous variables, as follows:

$$(BIL)$$

$$max\ \ c^t x + \sum_{j \notin J} {q^j}^t \theta x_j + \sum_{j \in J} {q^j}^t \delta^j,\ \ min\ \ p^t\theta\ s.t.$$

$$A_i x + \sum_{j \notin J} {Q^{ij}}^t \theta x_j + \sum_{j \in J} {Q^{ij}}^t \delta^j \leq b_i + {d^i}^t\theta\ \ \forall i$$

$$x_j\underline{\theta} \leq \delta^j \leq x_j\overline{\theta},\ \ (1 - x_j)\underline{\theta} \leq \theta - \delta^j \leq (1 - x_j)\overline{\theta}\ \ \forall j \in J$$

$$x \in \Omega,\ \ \underline{\theta} \leq \theta \leq \overline{\theta},\ \ \delta \in \Re^{r \times n}$$

Note that if $(x, \theta, \delta) \in F(BIL)$ then $\delta^j = x_j\theta\ \ \forall j \in J$ and $(x, \theta) \in F(BI)$. Also, if $(x, \theta) \in F(BI)$ then $(x, \theta, \delta) \in F(BIL)$ with $\delta^j = x_j\theta\ \ \forall j \in J$ and the equivalence follows immediately.

Let $(\hat{x}, \hat{\theta}, \hat{\delta}) \in F(BIL)$. Remember that $(\hat{x}, \hat{\theta}, \hat{\delta})$ is an efficient solution if and only if there is not another feasible $(x, \theta, \delta)$ such that $c^t x \geq c^t\hat{x}$, $p^t\hat{\theta} \geq p^t\theta$ with at least one strict inequality. The resultating criterion vector $(c(\hat{\theta})^t\hat{x}, p^t\hat{\theta})$ is said to be non-dominated. The pareto set of $BIL$ is the set of its non-dominated solutions. The pareto set of $BIL$ (even an approximation) is a valuable knowledge to the decision maker.

From the theoretical point of view we may use 0-1-MIBLP ([8]) to generate an approximation to the pareto set of $BIL$ by using the same ideas that work for 0-1-MILP problems ([16],[17]). If $q^j = 0\ \forall j \notin J$ and $Q^{ij} = 0\ \forall j \notin J, \forall i$ then $BIL$ is a biobjective 0-1-MILP problem and we can generate an approximation of the pareto set by using multiobjective 0-1-MILP ([16],[17]).

However, we do not have computational results for that general cases at this time. In the rest of the paper we restrict the analysis to 0-1-ILP problems with $q^j = 0 \ \forall j$.

# 3 The pareto set of $BIL$ for 0-1-Integer Linear Programming problems. Case: $q^j = 0 \ \forall j$.

Let us suppose that $P$ is a 0-1-ILP problem. In this section we present algorithms to obtain: (i) the exact pareto set for $BIL$ and, for practical purpose, (ii) an approximate pareto set for $BIL$.

Let us suppose that $q^j = 0 \ \forall j$. Therefore $P$, $P(\theta)$ and $BIL$ may be rewritten as follows:

$$(P) \ \ max \ \ c^t x \ \ s.t.$$
$$Ax \leq b$$
$$x \in \Omega = \{x : \hat{A}x \leq \hat{b}, \ \ x \in \{0,1\}^n\}$$

$$(P(\theta)) \ \ max \ \ c^t x \ \ s.t.$$
$$A_i x + \sum_{j=1}^{n} {Q^{ij}}^t \theta x_j \leq b_i + {d^i}^t \theta \ \ \forall i$$
$$x \in \Omega$$

$$BIL$$
$$max \ \ c^t x, \ \ min \ \ p^t \theta \ s.t.$$
$$A_i x + \sum_{j=1}^{n} {Q^{ij}}^t \delta^j \leq b_i + {d^i}^t \theta \ \ \forall i$$
$$x_j \underline{\theta} \leq \delta^j \leq x_j \overline{\theta}, \ \ (1 - x_j)\underline{\theta} \leq \theta - \delta^j \leq (1 - x_j)\overline{\theta} \ \ \forall j$$
$$x \in \Omega, \ \ \underline{\theta} \leq \theta \leq \overline{\theta}, \ \ \delta \in \Re^{r \times n}$$

Let $w \geq 0$. If the decision maker has limited funds $(w)$ the optimal investment may be obtained by solving the following problem in $(\theta)$:

$$(OI(w)) \;\; max \;\; v(P(\theta)) \;\; s.t.$$
$$p^t\theta \leq w$$
$$\underline{\theta} \leq \theta \leq \overline{\theta}$$

Since $\underline{\theta} \leq 0 \leq \overline{\theta}$ we have that $F(OI(w)) \neq \emptyset$.

$OI(w)$ may be rewritten as a 0-1-MILP problem in $(x, \theta, \delta)$ following the same standard linearization procedure:

$$(OIL(w)) \;\; max \;\; c^t x \;\; s.t.$$
$$A_i x + \sum_{j=1}^{n} {Q^{ij}}^t \delta^j \leq b_i + {d^i}^t \theta \;\; \forall i$$
$$x_j \underline{\theta} \leq \delta^j \leq x_j \overline{\theta}, \;\; (1 - x_j)\underline{\theta} \leq \theta - \delta^j \leq (1 - x_j)\overline{\theta} \;\; \forall j$$
$$p^t\theta \leq w$$
$$x \in \Omega, \;\; \underline{\theta} \leq \theta \leq \overline{\theta}, \;\; \delta \in \Re^{r \times n}$$

Let $g(w) = v(OI(w)) = v(OIL(w)) \quad \forall w \geq 0$. If $0 \leq w_1 \leq w_2$ then: $F(OIL(w_1)) \subseteq F(OIL(w_2))$ and then because of $OIL(w)$ is a 0-1-MILP problem $g$ is an upper semicontinuous, piecewise linear, nondecreasing and bounded function. Also, since the objetive function depends only on binary variables then $g$ is a step function. We can use parametric programming ([14]) in order to find $g(w)$ for all $w \geq 0$. Note that if we know $g(w)$ for all $w \geq 0$ then we know the pareto set of $BIL$.

## 3.1 The parametric algorithm.

$OIL(w)$ is a 0-1-MILP problem and is a special case of $G(w)$, a 0-1-MILP problem in $(x, y)$, defined as follows:

$$(G(w)) \;\; max \;\; c^t x \;\; s.t.$$
$$A^{1x}x + A^{1y}y \leq b^1 + wh$$
$$A^{2x}x + A^{2y}y \leq b^2$$
$$x \in \{0,1\}^n, \;\; y \geq 0$$

where $c$, $b^1$, $b^2$ and $0 \leq h$ are vectors with appropriate dimensions and $A^{1x}$, $A^{1y}$, $A^{2x}$ and $A^{2y}$ are matrices with appropriate dimensions.

Let us suppose that $F(G(0)) \neq \emptyset$ and $H = \{(x, y) : \;\; A^{2x}x + A^{2y}y \leq b^2, \;\; x \in \{0,1\}^n, \;\; y \geq 0\}$ is a bounded set.

Let $g(w) = v(G(w))$ for all $w \geq 0$. Note that if $0 \leq w_1 \leq w_2$ then $F(G(w_1)) \subseteq F(G(w_2))$ and again $g$ is a nondecreasing, uppersemicontinuous and bounded function. Also, $g$ is a step function.

From the theorethical point of view we may use the algorithm described in [4] in order to obtain $g$. However the special structure of $G(w)$ may be used go define a specialized algorithm following the ideas from [5].

Let $G(\infty)$ be a 0-1-MILP problem in $(x, y)$ defined as follows:

$$(G(\infty)) \;\; max \;\; c^t x \;\; s.t.$$
$$(x, y) \in H$$

Since $H$ is a bounded set and $\emptyset \neq F(G(0)) \subseteq F(G(\infty))$ then there exists an optimal solution for $G(\infty)$. Let $(x^1, y^1)$ be an optimal solution for $G(\infty)$.

Since $G(\infty)$ is a relaxation of $G(w)$ for all $w \geq 0$ then $g(w) = v(G(w)) \leq v(G(\infty)) = c^t x^1$ for all $w \geq 0$.

Let $\hat{w} \geq 0$ and let $(\hat{x}, \hat{y}) \in F(G(\hat{w}))$. Let $W(\hat{x})$ a problem in $(w, y)$ defined as follows:

$$W(\hat{x}) \;\; min \;\; w \;\; s.t \;\; (\hat{x}, y) \in F(G(w)), \;\; w \geq 0$$

Note that with $W(\hat{x})$ we are looking for the feasibility interval for $\hat{x}$. That is: if $w \in [v(W(\hat{x})), \infty)$ then there exits $y$ such that $(\hat{x}, y) \in F(G(w))$.

$W(\hat{x})$ may be rewritten as a Linear Programming (LP) problem in $(w, y)$ as follows:

$$(WL(\hat{x})) \;\; min \;\; w \;\; s.t.$$
$$A^{1x}\hat{x} + A^{1y}y \leq b^1 + wh$$
$$A^{2x}\hat{x} + A^{2y}y \leq b^2$$
$$y \geq 0, \;\; w \geq 0$$

Since $(\hat{x}, \hat{y}) \in F(G(\hat{w}))$ then $F(WL(\hat{x})) \neq \emptyset$ and there exists optimal solution because of $w \geq 0$.

Let $w_1 = v(W(x^1))$ then $x^1$ is an optimal solution for $G(w)$ for all $w \geq w_1$ and $g(w) = c^t x^1$ for all $w \geq w_1$.

If $w_1 = 0$ then we know $g$. Let us suppose that $w_1 > 0$. We need an auxiliary problem to perform a complete parametric analysis. Let $\hat{w} \geq 0$. Let $X(\hat{w}) = \{x : \exists \text{ y such that } (x, y) \in F(G(\hat{w})), \nexists \text{ y such that } (x, y) \in F(G(w)) \;\; \forall w \text{ such that } w < \hat{w}\}$. Note that $X(0) = F(G(0))$. Let $X \subseteq X(\hat{w})$. Let $R(\hat{w}, X)$ a problem in $(x, y)$ defined as follows:

$$(R(\hat{w}, X)) \;\; max \;\; c^t x \;\; s.t. \;\; (x, y) \in F(G(\hat{w})), \;\; x \notin X$$

$R(\hat{w}, X)$ may be rewritten as a 0-1-MILP problem in $(x, y)$ as follows:

$$(RL(\hat{w}, X)) \;\; max \;\; c^t x \;\; s.t.$$

$$A^{1x}x + A^{1y}y \leq b^1 + \hat{w}h$$

$$x \in H, \;\; x \notin X$$

Where $x \notin X$ may be rewritten as usual. If $\hat{x} \in X$ let $K1(\hat{x}) = \{j : \hat{x}_j = 1, \; j = 1, \cdots, n\}$ and let $K0(\hat{x}) = \{j : \hat{x}_j = 0, \; j = 1, \cdots, n\}$ then $x \notin X$ is replaced by:

$$\sum_{j \in K1(\hat{x})} x_j - \sum_{K0(\hat{x})} x_j \leq |K1(\hat{x})| - 1 \;\; \forall \hat{x} \in X$$

**Lemma 1** *Let* $0 < \hat{w}$ *and let* $X \subseteq X(\hat{w})$*. Therefore: (i)* $F(RL(\hat{w}, X)) \neq \emptyset$*, (ii) there exists an optimal solution for* $RL(\hat{w}, X)$ *and (iii) if* $(x^*, y^*)$ *is an optimal solution for* $RL(\hat{w}, X)$ *and* $v(W(x^*)) < \hat{w}$ *then* $g(w) = c^t x^* \;\; \forall w \in [v(W(x^*)), \hat{w})$

Proof:

(i) If $F(RL(\hat{w}, X)) = \emptyset$ then $F(G(w)) = \emptyset \;\; \forall w \in [0, \hat{w})$ and $X = X(\hat{w})$. Since $F(G(0)) \neq \emptyset$ it follows that $F(R(\hat{w}, X)) \neq \emptyset$.

(ii) Since $H$ is a bounded set it follows that there exists an optimal solution for $RL(\hat{w}, X)$.

(iii) Let $w \in [v(W(x^*)), \hat{w})$ then there exists $y$ such that $(x^*, y) \in F(G(w))$. Let $(x, y) \in F(G(w)) \subseteq F(G(\hat{w}))$. We have that $x \notin X(\hat{w})$ and then $x \notin X$. Therefore $(x, y) \in F(RL(\hat{w}, X))$ and $c^t x \leq c^t x^*$, hence $g(w) = v(RL(w)) = c^t x^*$.

From lemma 1 and because of $X(w)$ is a finite set for all $w \geq 0$ the following algorithm may be used to obtain $g$.

We use a standard branch and bound (branch and cut) algorithm and a standard simplex algorithm according the case to solve the problems that appear when the algorithm is executed.

**The parametric algorithm (PA1)**

Let $(x^1, y^1)$ be an optimal solution for $G(\infty)$. Let $w_1 = v(W(x^1))$. Let $X = \{x^1\}$ and $k = 1$.

**while** $w_k > 0$
  $w^* = w_k$
  **while** $w^* = w_k$
    Solve $RL(w_k, X)$. Let $(x^*, y^*)$ be an optimal solution.
    Solve $WL(x^*)$. Let $(w^*, y^*)$ be an optimal solution.
    **If** $w^* = w_k$
      $X = X \cup \{x^*\}$
    **else**
      $x^{k+1} = x^*$, $w_{k+1} = w^*$, $X = \{x^*\}$, $k = k + 1$
    **endif**
  **endwhile**
**endwhile**

Note that $g(w) = c^t x^1$ for all $w \in [w_1, \infty)$ and $g(w) = c^t x^{k+1}$ for all $w_{k+1} \leq w < w_k$ for all $k \geq 1$.

In practice solving $RL(w_k, X)$ to optimality again and again may be a very expensive computational task. In practice we may use a Branch and bound (Branch and cut) algorithm with a tolerance (either a relative tolerance or an absolute tolerance) as follows: instead to solve $RL(w_k, X)$ to optimality we obtain $(x, y) \in F(RL(w_k, X))$ such that $(x, y)$ is an $\epsilon$-optimal solution (either $c^t x \leq v(RL(w_k, X)) \leq c^t x(1 + \epsilon)$ or $c^t x \leq v(RL(w_k, X)) \leq c^t x + \epsilon$ according to the tolerance used). The approximate algorithm is defined if we use an $\epsilon$-optimal solution of $RL(w_k, X)$ instead of an optimal solution.

Note that the $\epsilon$-optimality is valid for all $w \geq 0$ because of:

(i) If $w \geq w_1$ then either $c^t x^1 \leq g(w) \leq c^t x^1 (1+\epsilon)$ or $c^t x^1 \leq g(w) \leq c^t x^1 + \epsilon$ according to the tolerance used and

(ii) Let $k \geq 1$. Let $w \in [w_{k+1}, w_k)$. Let $jmax$ be the index such that $max\{c^t x^j : w_j \leq w,\ j \geq k+1\} = c^t x^{jmax}$ then either $c^t x^{k+1} \leq c^t x^{jmax} \leq g(w) \leq c^t x^{k+1}(1+\epsilon)$ or $c^t x^{k+1} \leq c^t x^{jmax} \leq g(w) \leq c^t x^{k+1} + \epsilon$ according to the tolerance used.

According to our computational results the parametric algorithm may be an expensive task even if we use a tolerance. In order to save computational efforts we design another $\epsilon$-optimal parametric algorithm to be presented in the next subsection.

## 3.2 The $\epsilon$-optimal parametric algorithm

We use a standard branch and bound (branch and cut) algorithm to obtain near optimal solutions according to the tolerance used.

### 3.2.1 The algorithm by using an absolute tolerance ($\epsilon, \epsilon/2$-PA2)

We use $\epsilon > 0$ as an absolute tolerance to obtain an approximation to $g$ and $\epsilon/2$ as an absolute tolerance to solve either $G(w)$ or $RL(w, X)$ for all $w$ and for all $X$.

We need to pointed out some remarks to present the algorithm.

**Remark 1** *Let* $w_1, w_2$ *with* $0 \leq w_1 < w_2$. *Let* $(x^1, y^1) \in F(G(w_1))$ *and let* $B$ *such that* $v(G(w_2)) \leq B$. *If* $B \leq c^t x^1 + \epsilon$ *then* $c^t x^1 \leq g(w) \leq c^t x^1 + \epsilon$ *for all* $w \in [w_1, w_2]$.

**Remark 2** *Let* $w_1, w_2$ *with* $0 \leq w_1 < w_2$. *Let* $(x^1, y^1) \in F(G(w_1))$ *and let* $B$ *such that* $v(R(w_2, X(w_2))) \leq B$. *If* $B \leq c^t x^1 + \epsilon$ *then* $c^t x^1 \leq g(w) \leq c^t x^1 + \epsilon$ *for all* $w \in [w_1, w_2)$.

**Remark 3** *Let* $\hat{w} > 0$. *Let* $(x^*, y^*)$ *be an* $\epsilon/2$*-optimal solution for* $R(w, X(\hat{w}))$ *then* $c^t x^* \leq g(w) \leq c^t x^* + \epsilon/2$ *for all* $w \in [v(W(x^*)), \hat{w})$

**Remark 4** *Let* $w_1, w_2$ *with* $0 \leq w_1 < w_2$. *Let* $(x^1, y^1)$ *be an* $\epsilon/2$*-optimal solution for* $G(w_1)$. *Let* $(x^2, y^2) \in F(G(w_2))$. *If* $v(W(x^2)) \leq w_1$ *then* $c^t x^2 \leq c^t x^1 + \epsilon/2$

From lemma 1, remarks 1,2,3 and 4 and because of $X(w)$ is a finite set for all $w \geq 0$ the following algorithm is well defined and may be used to obtain an $\epsilon$-approximation to $g$.

**The $\epsilon$-optimal parametric algorithm**

Let $(x^1, y^1)$ be an $\epsilon/2$-optimal solution for $G(0)$ and let $w_1 = 0$. Let $(x^2, y^2)$ be an $\epsilon/2$-optimal solution for $G(\infty)$. Let $w_2 = v(W(x^2))$. Let $z^1, z^2$ be the bounds obtained when we solve $G(0)$ and $G(\infty)$ approximately ($c^t x^1 \leq v(G(0)) \leq c^t x^1 + z^1 \leq c^t x^1 + \epsilon/2$ and $c^t x^2 \leq v(G(\infty)) \leq c^t x^2 + z^2 \leq c^t x^2 + \epsilon/2$). Let $k = 1$ and $s_1 = 1$.

**while** $\sum_{j=1}^{k} s_j > 0$

Select $r$ such that $z^{r+1} - c^t x^r = max\{s_j(z^{j+1} - c^t x^j) : \ j = 1, \cdots, k\}$.

**If** $z^{r+1} - c^t x^r \leq \epsilon$

$s_r = 0$

**else**

$X = \{x^{r+1}\}$, $w^* = w_{r+1}$.

**while** $w^* = w_{r+1}$

Let $(x^*, y^*)$ be an $\epsilon/2$-optimal solution for $RL(w_{r+1}, X)$ and let $z^*$ be the bound obtained.

Solve $WL(x^*)$. Let $(w^*, y^*)$ be an optimal solution.

**If** $w^* = w_{r+1}$ **then** $X = X \cup \{x^*\}$.

**endwhile**

**If** $z^* - c^t x^r \leq \epsilon$

$s_r = 0$

**else**

**Insert** $(w^*, c^t x^*, z^*, x^*)$ **with index** $r + 1$:

$(w_{j+1}, c^t x^{j+1}, z^{j+1}, x^{j+1}, s_{j+1}) = (w_j, c^t x^j, z^j, x^j, s_j)$ from $j = k + 2$ until $j = r + 2$.

$(w_{r+1}, c^t x^{r+1}, z^{r+1}, x^{r+1}) = (max\{w^*, w_r\}, c^t x^*, z^*, x^*)$.

**endinsert**

$k = k + 1$, $s_{r+1} = 0$.

$w = w_r + 0.5(w_{r+1} - w_r)$.

Let $(x^*, y^*)$ be an $\epsilon/2$-optimal solution for $G(w)$ and let $z^*$ be the bound obtained.

Solve $WL(x^*)$. Let $(w^*, y^*)$ be an optimal solution.

$sleft = 1$, $sright = 1$.

**If** $z^{r+1} - c^t x^* \leq \epsilon$ **then** $ssright = 0$

**If** $z^* - c^t x^r \leq \epsilon$ **then** $sleft = 0$

**Insert** $(w^*, c^t x^*, z^*, x^*)$ **with index** $r + 1$

**endinsert**

$s_r = sleft$, $s_{r+1} = sright$, $k = k + 1$

**endif**

**endif**

**endwhile**

The algorithm works as follows: first we solve $G(0)$ and $G(\infty)$ to obtain $x^1, x^2, z^2$ and let $w_2 = v(W(x^2))$. Now $k = 1$ and because of $s_1 = 1$ we select $r = 1$.

If $z^2 - c^t x^1 < \epsilon$ then because of remark 1 we have $c^t x^1 \leq g(w) \leq c^t x^1 + \epsilon$ for all $w \in [w_1, w_2] = [0, v(W(x^2)]$ and since $c^t x^2 \leq g(w) = g(w_2) \leq c^t x^2 + \epsilon/2$ for all $w \geq w_2$ the parametric analysis is complete ($s_1 = 0$) and the algorithm stops.

If $z^2 - c^t x^1 > \epsilon$ the interval $[w_1, w_2) = [0, v(W(x^2)))$ must be evaluated ($s_1 = 1$).

Next we obtain $(x^*, y^*)$, an $\epsilon/2$-optimal solution for $RL(w_2, X(w_2))$, and $z^*$ such that $v(RL(w_2, X(w_2))) \leq z^*$.

If $z^* - c^t x^1 \leq \epsilon$ then because of remark 2 we have $c^t x^1 \leq g(w) \leq c^t x^1 + \epsilon$ for all $w \in [w_1, w_2)$ and since $c^t x^2 \leq g(w) = g(w_2) \leq c^t x^2 + \epsilon/2$ for all $w \geq w_2$ the parametric analysis is complete ($s_1 = 0$) and the algorithm stops.

If $z^* - c^t x^1 > \epsilon$ we insert $(w^*, c^t x^*, z^*, x^*)$ as follows: $(w_3, c^t x^3, z^3, x^3, s_3) = (w_2, c^t x^2, z^2, x^2, s_2)$, $(w_2, c^t x^2, z^2, x^2) = (max\{w^*, w_2\}, c^t x^*, z^*, x^*)$ and because of remark 3 we have $c^t x^2 \leq g(w) \leq c^t x^2 + \epsilon/2$ for all $w \in [w_2, w_3)$ and $s_2 = 0$. Now we have two intervals with $s_1 = 1$ and $s_2 = 0$.

Now let $(x^*, y^*)$ be an $\epsilon/2$-optimal solution for $G(w_1 + 0.5(w_2 - w_1))$ and let $z^*$ the bound obtained. Next we solve $WL(x^*)$ to obtain $(w^*, y^*)$. Now we insert $(w^*, c^t x^*, z^*, x^*)$ and we have three intervals and according to the value of $sleft$ and $sright$ we have four scenarios as follows:

$(s_1, s_2, s_3) \in \{(0,0,0), (1,0,0), (0,1,0), (1,1,0)\}$.

In the first case the algorithm stops. In the other cases the algorithm continues by choosing the first or the second interval according the rule and so on.

When the algorithm stops we have:

(i) If $w \geq w_k$ then $c^t x^k \leq g(w) \leq c^t x^k + \epsilon/2$.

(ii) Let $j \leq k-1$. Let $w \in [w_j, w_{j+1})$ then $c^t x^j \leq g(w) \leq c^t x^j + \epsilon$.

(iii) If $w = w_j$ then $c^t x^j \leq g(w) \leq c^t x^j + \epsilon/2$ for all $j$.

Any rule to select the next interval may be used. We use the rule presented since appears to be an appropriate choice if we stop the algorithm by using a time limit.

Note that because of remark 4 if $v(W(x^*)) \leq w_r$ then $z^* \leq c^t x^* + \epsilon/2 \leq c^t x^r + \epsilon$ and then if $(w^*, c^t x^*, z^*, x^*)$ is inserted the interval $r$ will not be selected again ($s_r = sleft = 0$) and then the degenerate intervals will not be a problem neither from the theoretical nor from the practical point of view.

### 3.2.2 The algorithm by using a relative tolerance ($\epsilon, \epsilon_1$-PA3)

We use $\epsilon_1 > 0$ as a relative tolerance to solve either $G(w)$ or $RL(w, X)$ for all $w$ and for all $X$ and $\epsilon = 2\epsilon_1 + \epsilon_1^2$ as a relative tolerance to obtain an approximation to $g$ .

The remarks 1 to 4 may be rewritten by using a relative tolerance as follows:

**Remark 5** *Let $w_1, w_2$ with $0 \leq w_1 < w_2$. Let $(x^1, y^1) \in F(G(w_1))$ and let $B$ such that $v(G(w_2)) \leq B$. If $B \leq c^t x^1(1+\epsilon)$ then $c^t x^1 \leq g(w) \leq c^t x^1(1+\epsilon)$ for all $w \in [w_1, w_2]$.*

**Remark 6** *Let $w_1, w_2$ with $0 \leq w_1 < w_2$. Let $(x^1, y^1) \in F(G(w_1))$ and let $B$ such that $v(R(w_2, X(w_2))) \leq B$. If $B \leq c^t x^1(1+\epsilon)$ then $c^t x^1 \leq g(w) \leq c^t x^1(1+\epsilon)$ for all $w \in [w_1, w_2)$.*

**Remark 7** *Let $\hat{w} > 0$. Let $(x^*, y^*)$ be an $\epsilon_1$-optimal solution for $R(\hat{w}, X(\hat{w}))$ then $c^t x^* \leq g(w) \leq c^t x^*(1+\epsilon_1)$ for all $w \in [v(W(x^*)), \hat{w})$*

**Remark 8** *Let Let $w_1, w_2$ with $0 \leq w_1 < w_2$. Let $(x^1, y^1)$ be an $\epsilon_1$-optimal solution for $G(w_1)$. Let $(x^2, y^2) \in F(G(w_2))$. If $v(W(x^2)) \leq w_1$ then $c^t x^2 \leq c^t x^1(1+\epsilon_1)$*

Let us suppose that $c > 0$ and $(x, y) = (0, y)$ is not an $\epsilon_1$-optimal solution for $G(w)$ for all $w$ and for all $y$.

From lemma 1, remarks 5,6,7 and 8 and because of $X(w)$ is a finite set for all $w \geq 0$ the algorithm by using a relative tolerance is well defined and may be used to obtain an $\epsilon$-approximation to $g$.

The algorithm may be rewritten to use relative tolerance as follows:

(i) Let $(x^1, y^1)$ be an $\epsilon_1$-optimal solution for $G(0)$ and let $w_1 = 0$. Let $(x^2, y^2)$ be an $\epsilon_1$-optimal solution for $G(\infty)$. Let $w_2 = v(W(x^2))$. Let $z^1, z^2$ be the bounds obtained when we solve $G(0)$ and $G(\infty)$ approximately by using a standard branch and bound ( and cut) algorithm ($c^t x^1 \leq v(G(0)) \leq c^t x^1 + z^1 \leq c^t x^1(1+\epsilon_1)$ and $c^t x^2 \leq v(G(\infty)) \leq c^t x^2 + z^2 \leq c^t x^2(1+\epsilon_1)$). Let $k = 1$ and $s_1 = 1$,

(ii) replace Select $r$ such that $z^{r+1} - c^t x^r = max\{s_j(z^{j+1} - c^t x^j) : j = 1, \cdots, k\}$ by Select $r$ such that $\frac{(z^{r+1} - c^t x^r)}{c^t x^r} = max\{\frac{s_j(z^{j+1} - c^t x^j)}{c^t x^j} : j = 1, \cdots, k\}$ .

(iii) replace **If** $z^{r+1} - c^t x^r \leq \epsilon$ by **If** $\frac{z^{r+1} - c^t x^r}{c^t x^r} \leq \epsilon$,

(iv) replace **If** $z^{r+1} - c^t x^* \leq \epsilon$ by **If** $\frac{z^{r+1} - c^t x^*}{c^t x^*} \leq \epsilon$ and

(v) replace **If** $z^* - c^t x^r \leq \epsilon$ by **If** $\frac{z^* - c^t x^r}{c^t x^r} \leq \epsilon$

When the algorithm stops we have:

(i) If $w \geq w_k$ then $c^t x^k \leq g(w) \leq c^t x^k(1+\epsilon_1)$.

(ii) Let $j \leq k-1$. Let $w \in [w_j, w_{j+1})$ then $c^t x^j \leq g(w) \leq c^t x^j(1+\epsilon)$.

(iii) If $w = w_j$ then $c^t x^j \leq g(w) \leq c^t x^j(1+\epsilon_1)$ for all $j$.

We use $\epsilon = 2\epsilon_1 + \epsilon_1^2$ to be sure that if $v(W(x^*)) \leq w_r$ then $\frac{z^* - c^t x^r}{c^t x^r} \leq \epsilon$ and then if $(w^*, c^t x^*, z^*, x^*)$ is inserted the interval $r$ will not be selected again ($s_r = sleft = 0$).

# 4 Computational results

In this section we present computational results by using the parametric and the $\epsilon$-optimal parametric algorithms presented to obtain an approximation of $g$ (and as a consequence an approximation of the pareto set for $BI$).

The problems considered are: (i) Multidimensional Knapsack (KPm) problems ([6],[15]) and (ii) Bounded Knapsack (BKP) problems ([11]).

Because of space considerations we present results about the PA1 algorithm with relative tolerance for KPm problems, the PA1 algorithm with absolute tolerance for BKP problems and the $\epsilon, \epsilon/2$-PA2 algorithm for BKP problems. The results by using the $\epsilon, \epsilon_1$-PA2 algorithm (with relative tolerance) are analogous.

## 4.1 Multidimensional Knapsack Problems

A set of $n$ items with value $c_j$ $(j = 1, \cdots, n)$ and $m + s$ resources with capacities $b_i$ $(i = 1, \cdots, m)$ and $\hat{b}_i$ $(i = 1, \cdots, s)$ are given. Each item consumes an amount from each resource ($A_{ij}$ and $\hat{A}_{ij}$). The 0-1 decision variables $x_j$ indicate which items are selected. The objective is to choose a subset of items with maximum total value. Selected items must, however, not exceed resource capacities.

The formulation of a Multidimensional Knapsack (KPm) problem is:

$$(P) \;\; max \;\; c^t x \;\; s.t.$$

$$Ax \leq b$$

$$x \in \Omega$$

where $\Omega = \{x : \hat{A}x \leq \hat{b},\ x \in \{0,1\}^n\}$ and $0 < b \in \Re^m,\ 0 < \hat{b} \in \Re^s,\ 0 \leq A \in \Re^{m\times n},\ 0 \leq \hat{A} \in \Re^{s\times n}$.

Let us suppose that the decision maker may buy additional units of the resources and may pay in order to reduce the value of the coefficients of the matrix. All modifications are mutually independent. We need $mn + m$ parameters as follows: $\theta_{(i-1)n+j}$ to define the change in the coefficient $A_{ij}$ $(i = 1, \cdots, m)(j = 1, \cdots, n)$ and $\theta_{mn+i}$ to define the change in the availability of resource $i$ $(i = 1, \cdots, m)$.

Let $\overline{\theta} \in [0,1]^{mn+n}$. Let $\theta \in [0, \overline{\theta}]$. The problem $P(\theta)$ considered is:

$$(P(\theta))\ \ max\ \ c^t x\ \ s.t.$$

$$A_i x - \sum_{j=1}^{n} A_{ij}\theta_{(i-1)n+j}x_j \leq b_i(1+\theta_{nm+i})\ \ \forall i$$

$$x \in \Omega$$

Let $p^t\theta$ the price of the modification with $p \in \Re^{nm+m}$.

The $BI$ problem is:

$$(BI)$$

$$max\ \ c^t x,\ \ min\ \ p^t\theta\ \ s.t.$$

$$A_i x - \sum_{j=1}^{n} A_{ij}\theta_{(i-1)n+j}x_j \leq b_i(1+\theta_{nm+i})\ \ \forall i$$

$$x \in \Omega,\ \ 0 \leq \theta \leq \overline{\theta}$$

Since all data and its modifications are mutually independent we do not need to use the variable vector $\delta$ and $BI$ may be rewritten as follows:

$$(BIL)$$

$$max\ \ c^t x,\ \ min\ \ p^t\theta\ s.t.$$

$$A_i x - \sum_{j=1}^{n} A_{ij}\theta_{(i-1)n+j} \leq b_i(1+\theta_{nm+i})\ \ \forall i$$

$$0 \leq \theta_{(i-1)n+j} \leq x_j\overline{\theta}_{(i-1)n+j},\ \ 0 \leq \theta_{mn+i} \leq \overline{\theta}_{mn+i}\ \ \forall i,\ \forall j$$

$$x \in \Omega,\ \ 0 \leq \theta \leq \overline{\theta}$$

The $OI(w)$ problem is:

$$(OI(w))\ \ max\ \ c^t x\ \ s.t.$$

$$A_i x - \sum_{j=1}^{n} A_{ij}\theta_{(i-1)n+j} x_j \leq b_i(1+\theta_{nm+i})\ \ \forall i$$

$$p^t\theta \leq w$$

$$x \in \Omega,\ \ 0 \leq \theta \leq \overline{\theta}$$

and $OIL(w)$ is:

$$(OIL(w))\ \ c^t x\ \ s.t.$$

$$A_i x - \sum_{j=1}^{n} A_{ij}\theta_{(i-1)n+j} \leq b_i(1+\theta_{nm+i})\ \ \forall i$$

$$0 \leq \theta_{(i-1)n+j} \leq x_j\overline{\theta}_{(i-1)n+j},\ \ 0 \leq \theta_{mn+i} \leq \overline{\theta}_{mn+i}\ \ \forall i,\ \forall j$$

$$p^t\theta \leq w$$

$$x \in \Omega,\ \ 0 \leq \theta \leq \overline{\theta}$$

The original data were generated at random following standard procedures as follows $(j = 1, \cdots, n)$ $(i = 1, \cdots, m)(k = 1, \cdots, s)$:

$A_{ij}$ is taken from $U(1, 1000)$ and $\hat{A}_{kj}$ is taken from $U(1, 1000)$. $c_j = 500u_j + (\sum_{i=1}^{m} A_{ij} + \sum_{i=1}^{s} \hat{A}_{ij})/(m+s)$ and $u_j$ is taken from $U(0, 1)$. $b_i = 0.5\sum_{j=1}^{k} A_{ij}$ and $\hat{b}_i = 0.5\sum_{j=1}^{k} \hat{A}_{ij}$.

Let $\lambda \in [0, 1]$. Let $\underline{\theta} = 0$ and let $\overline{\theta}$ be the upper bound of $\theta$ generated as follows:

$\overline{\theta}_{(i-1)n+j} = \lambda u_{ij}$ and $u_{ij}$ is taken from $U(0, 1)$.

$\overline{\theta}_{nm+i} = \lambda u_i$ and $u_i$ is taken from $U(0, 1)$.

The price of the modification is $p^t\theta$ where $p \in \Re^{nm+m}$ is generated as follows $(i = 1, \cdots, m)(j = 1, \cdots, n)$:

$p_{(i-1)n+j} = u_{ij}$ and $u_{ij}$ is taken from $U(0, 1)$,

$p_{nm+i} = u_i$ and $u_i$ taken from $U(0, 1)$.

We use the parametric algorithm (PA1) to obtain an $\epsilon$-approximation to $g$ by using relative tolerance . For each $(n, m, s, \lambda, \epsilon)$ considered we generate 10 problems and report: the minimal ($mint$), average ($\overline{t}$) and maximal time ($maxt$) in seconds to perform the complete parametrical analysis, the minimal ($mink$), average ($\overline{k}$) and maximal ($maxk$) number of solutions generated, and the minimal ($mininc$), average ($\overline{inc}$) and maximal ($maxinc$) porcentual increase of the original objective function with no limit to the funds ($100 \times \frac{v(OI(\infty))-v(P(0))}{v(P(0))}$).

In Table 1 we present the results with $n = 1000$ and $\epsilon = 0.001$ to define the relative tolerance. If $m + s = 20$ then we use $\epsilon = 0.00125$.

| $m,s$ | $\lambda$ | $mint$ | $\overline{t}$ | $maxt$ | $mink$ | $\overline{k}$ | $maxk$ | $mininc$ | $\overline{inc}$ | $maxinc$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1,9 | 0.50 | 20.88 | 35.67 | 57.34 | 55 | 86.70 | 131 | 0.28 | 0.45 | 0.75 |
| | 0.10 | 24.57 | 32.50 | 44.57 | 50 | 80.50 | 108 | 0.30 | 0.47 | 0.59 |
| 5,5 | 0.50 | 111.97 | 135.26 | 146.40 | 236 | 253.30 | 276 | 3.37 | 3.87 | 4.36 |
| | 0.10 | 73.59 | 111.78 | 198.52 | 128 | 147.50 | 170 | 2.73 | 2.94 | 3.25 |
| 9,1 | 0.50 | 454.36 | 567.49 | 629.45 | 557 | 598.40 | 623 | 20.55 | 21.45 | 22.48 |
| | 0.10 | 212.95 | 255.40 | 323.14 | 163 | 196.90 | 230 | 6.08 | 7.39 | 8.98 |
| 1,11 | 0.50 | 38.79 | 51.83 | 69.26 | 53 | 69.80 | 106 | 0.25 | 0.30 | 0.41 |
| | 0.10 | 29.27 | 52.89 | 77.29 | 50 | 69.20 | 86 | 0.20 | 0.36 | 0.46 |
| 6,6 | 0.50 | 161.05 | 182.93 | 199.31 | 217 | 234.40 | 247 | 3.54 | 3.82 | 4.22 |
| | 0.10 | 143.83 | 175.93 | 226.98 | 155 | 172.40 | 190 | 2.80 | 3.02 | 3.26 |
| 11,1 | 0.50 | 793.80 | 942.60 | 1039.00 | 603 | 638.40 | 685 | 21.41 | 22.11 | 23.30 |
| | 0.10 | 309.08 | 401.58 | 492.24 | 181 | 200.20 | 222 | 6.27 | 7.36 | 7.96 |
| 1,13 | 0.50 | 87.43 | 126.12 | 179.08 | 40 | 47.40 | 65 | 0.12 | 0.17 | 0.23 |
| | 0.10 | 102.41 | 165.54 | 237.98 | 40 | 51.20 | 81 | 0.10 | 0.22 | 0.37 |
| 7,7 | 0.50 | 224.89 | 311.70 | 410.79 | 211 | 224.20 | 236 | 2.83 | 2.91 | 3.03 |
| | 0.10 | 271.08 | 310.09 | 383.89 | 140 | 165.80 | 182 | 2.03 | 2.64 | 3.05 |
| 13,1 | 0.50 | 1199.01 | 1298.89 | 1442.93 | 615 | 647.40 | 679 | 21.52 | 22.13 | 22.39 |
| | 0.10 | 676.16 | 752.95 | 873.77 | 191 | 213.80 | 242 | 7.19 | 7.65 | 8.45 |
| 1,19 | 0.50 | 51.47 | 280.49 | 476.52 | 12 | 38.80 | 62 | 0.04 | 0.18 | 0.38 |
| | 0.10 | 77.53 | 305.33 | 551.78 | 13 | 30.80 | 55 | -0.00 | 0.12 | 0.28 |
| 10,10 | 0.50 | 366.19 | 593.38 | 1363.62 | 114 | 132.80 | 143 | 1.99 | 2.33 | 2.52 |
| | 0.10 | 473.53 | 554.15 | 727.88 | 111 | 120.20 | 133 | 1.68 | 2.11 | 2.39 |
| 19,1 | 0.10 | 1450.21 | 1923.65 | 2603.48 | 167 | 178.00 | 187 | 7.03 | 7.46 | 8.25 |

Table 1 PA1 algorithm. $n = 1000, \epsilon = 0.001$. If $m + s = 20$ then $\epsilon = 0.00125$

In Table 2 we present the results with $n = 1500$ and $\epsilon = 0.001$ to define the relative tolerance. If $m + s = 20$ then we use $\epsilon \in \{0.01, 0.005\}$.

| $m,s$ | $\lambda$ | $mint$ | $\overline{t}$ | $maxt$ | $mink$ | $\overline{k}$ | $maxk$ | $mininc$ | $\overline{inc}$ | $maxinc$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 9,1 | 0.10 | 284.97 | 367.12 | 497.22 | 134 | 175.80 | 231 | 5.94 | 7.12 | 8.89 |
| | 0.50 | 820.69 | 960.15 | 1055.93 | 514 | 567.70 | 610 | 20.42 | 21.34 | 22.64 |
| 19,1 | 0.10 | 931.94 | 1379.28 | 2251.56 | 78 | 104.70 | 123 | 6.80 | 7.60 | 8.68 |
| | | 883.18 | 1342.54 | 1764.70 | 76 | 105.40 | 123 | 6.76 | 7.33 | 8.10 |
| | 0.50 | 2012.92 | 2113.45 | 2213.99 | 387 | 388.00 | 389 | 22.44 | 23.04 | 23.64 |
| | | 2248.74 | 2354.11 | 2531.59 | 383 | 410.00 | 439 | 22.32 | 22.91 | 23.22 |

Table 2 PA1 algorithm. $n = 1500, \epsilon = 0.001$. If $m + s = 20$ then $\epsilon \in \{0.01, 0.005\}$

## 4.2 Bounded Knapsack Problems

The Bounded Knapsack (BKP) problem is defined as follows. Given $n$ items types and a knapsack with: $c_j$ the value and $w_j$ the weight af an item of type $j$, $F_j$ the availability of items of type $j$ and $W$ the capacity of the knapsack, the problem is select $x_j$ ($j = 1, \cdots, n$), the number of items of each type, in such a manner that the total value is maximized subject to the capacity of the knapsack and the availabilities of the items.

The formulation of a BKP problem is:

$$(P)\ max\ \ c^t x\ \ s.t.$$
$$w^t x \leq W$$
$$x \in \Omega$$

where $\Omega = \{x \in \Re^n,\ \ 0 \leq x \leq F,\ \ x\ \text{ integer}\}$

Let us suppose that the decision maker may pay in order to reduce the weight of the items. All modifications are mutually independents. We need $n$ parameters as follows: $\theta_j$ to define the change in the coefficient $w_j$ $(j = 1, \cdots, n)$.

Let $\overline{\theta} \in [0,1]^n$. Let $\theta \in [0, \overline{\theta}]$. The problem $P(\theta)$ considered is:

$$(P(\theta))\ \ max\ \ c^t x\ \ s.t.$$
$$\sum_{j=1}^{n} (w_j - \theta_j w_j) x_j \leq W$$
$$x \in \Omega$$

Let $p^t\theta$ the price of the modification with $p \in \Re^n$.

The $BI$ problem is:

$$(BI)$$
$$max\ \ c^t x,\ \ min\ \ p^t\theta\ \ s.t.$$
$$\sum_{j=1}^{n} (w_j - \theta_j w_j) x_j \leq W$$
$$x \in \Omega,\ 0 \leq \theta \leq \overline{\theta}$$

We may use some classical approaches [11] in order to rewrite $P$ as 0-1-ILP problem. However the classical transformation, designed to minimize the number of 0-1 variables, permit redundancies (a solution admit several representations in the 0-1 problem). Multiple redundancies may be a problem to the parametric algorithm. Since the same solution admits several representations we can obtain $w^* = w_k$ again and again when we are solving $RL(w_k, X)$. Thus, we need a transformation without redundancies as follows:

$$x_j = \sum_{k=1}^{F_j} k x_k^j,\ \ \sum_{k=1}^{F_j} x_k^j \leq 1,\ \ x_k^j \in \{0,1\}\ (j = 1, \cdots, n)(k = 1, \cdots, F_j)$$

With this transformation each solution has an unique representation in the 0-1 problem. Thus, $P$, $P(\theta)$ and $BI$ may be rewritten as follows:

$$(P) \;\; max \;\; \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j \;\; s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j x_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \;\; (j = 1, \cdots, n)$$

$$x_k^j \in \{0,1\} \;\; (j = 1, \cdots, n)(k = 1, \cdots, F_j)$$

$$(P(\theta)) \;\; max \;\; \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j \;\; s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} k(w_j - \theta_j w_j) x_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \;\; (j = 1, \cdots, n)$$

$$x_k^j \in \{0,1\} \;\; (j = 1, \cdots, n)(k = 1, \cdots, F_j)$$

$$(BI)$$

$$max \;\; \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j, \;\; min \;\; p^t\theta \;\; s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} k(w_j - \theta_j w_j) x_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \;\; (j = 1, \cdots, n)$$

$$0 \leq \theta \leq \overline{\theta}$$

$$x_k^j \in \{0,1\} \;\; (j = 1, \cdots, n)(k = 1, \cdots, F_j), \;\; \theta \in \Re^n$$

Althought the modifications considered are mutually independent for the original problem that is not the case in the 0-1 version of $(BI)$ because the value of $\theta_j$ affects the $F_j$ variables used to represent $x_j$. Thus, we need the $\delta$

variables to rewrite $BI$ as follows:

$$(BIL)$$

$$max \ \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j, \ \ min \ \ p^t\theta \ \ s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j x_k^j - \sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j \delta_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \ \ (j=1,\cdots,n)$$

$$0 \leq \delta_k^j \leq \overline{\theta}_j x_k^j \ \ (j=1,\cdots,n)(k=1,\cdots,F_j)$$

$$0 \leq \theta_j - \delta_k^j \leq \overline{\theta}_j (1-x_j^k) \ \ (j=1,\cdots,n)(k=1,\cdots,F_j)$$

$$0 \leq \theta \leq \overline{\theta}$$

$$x_k^j \in \{0,1\} \ \ (j=1,\cdots,n)(k=1,\cdots,F_j), \ \ \theta \in \Re^n, \ \delta^j \in \Re^{F_j}$$

Since $\sum_{k=1}^{F_j} x_k^j \leq 1 \ \ (j=1,\cdots,n)$ we have $\theta_j = \sum_{k=1}^{F_j} \delta_k^j x_k^j$ and we may delete the constraints $0 \leq \theta_j - \delta_k^j \leq \overline{\theta}_j(1-x_j^k)$ to rewrite $(BIL)$ as follows:

$$(BIL)$$

$$max \ \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j, \ \ min \ \ \sum_{j=1}^{n}\sum_{k=1}^{F_j} p_j \delta_k^j \ \ s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j x_k^j - \sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j \delta_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \ \ (j=1,\cdots,n)$$

$$0 \leq \delta_k^j \leq \overline{\theta}_j x_k^j \ \ (j=1,\cdots,n)(k=1,\cdots,F_j)$$

$$x_k^j \in \{0,1\} \ \ (j=1,\cdots,n)(k=1,\cdots,F_j), \ \ \delta^j \in \Re^{F_j} \ \ (j=1,\cdots,n)$$

and $OIL(w)$ is:

$$(OIL(w)) \ \ max \ \ \sum_{j=1}^{n}\sum_{k=1}^{F_j} kc_j x_k^j \ \ s.t.$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j x_k^j - \sum_{j=1}^{n}\sum_{k=1}^{F_j} kw_j \delta_k^j \leq W$$

$$\sum_{k=1}^{F_j} x_k^j \leq 1 \ \ (j = 1, \cdots, n)$$

$$0 \leq \delta_k^j \leq \overline{\theta}_j x_k^j \ \ (j = 1, \cdots, n)(k = 1, \cdots, F_j)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{F_j} p_j \delta_k^j \leq w$$

$$x_k^j \in \{0,1\} \ \ (j = 1, \cdots, n)(k = 1, \cdots, F_j), \ \ \delta^j \in \Re^{F_j} \ \ (j = 1, \cdots, n)$$

The original data were generated at random following standard procedures as follows ($j = 1, \cdots, n$):

$w_j$ is taken from $U(1, 1000)$, $W = 0.5 \sum_{j=1}^{n} F_j w_j$.

$F_j$ is taken from $U(6, 10)$.

Let $\lambda \in [0, 1]$. Let $\underline{\theta} = 0$ and let $\overline{\theta}$ be the upper bound of $\theta$ defined as follows:

$\overline{\theta}_j = \lambda u_j$ and $u_j$ is taken from $U(0, 1)$.

The price of the modification is $p^t\theta$ where $p \in \Re^n$ is generated as follows ($j = 1, \cdots, n$):

$p_j = u_j$ and $u_j$ is taken from $U(0, 1)$,

#### 4.2.1 The general case: $c_j$ is generated at random $\forall j$

We generate weakly correlated problems as follows: $c_j$ is taken from $U(w_j - 100, w_j + 100)$.

We use the parametric algorithm (PA1) and the $\epsilon$-optimal parametric algorithm ($\epsilon, \epsilon/2$-PA2) to obtain an $\epsilon$-approximation to $g$ with absolute tolerance. Let $\delta \in (0, 1)$ then we use $\epsilon = \delta v(P(0))$. For each $(n, \lambda, \delta)$ considered we generate 10 problems and report: the minimal ($mint$), average ($\overline{t}$) and maximal time ($maxt$) in seconds to perform the complete parametrical analysis, the minimal ($mink$), average ($\overline{k}$) and maximal ($maxk$) number of solutions generated, and the minimal ($mininc$), average ($\overline{inc}$) and maximal ($maxinc$)

porcentual increase of the original objective function with no limit to the funds $(100 \times \frac{v(OI(\infty))-v(P(0))}{v(P(0))})$.

In Table 3 we present the results. For each $n$ the first row corresponds to the PA1 algorithm and the second row corresponds to the $\epsilon, /epsilon/2$-PA2 algorithm.

| $n$ | $\lambda$ | $\delta$ | $mininc$ | $\overline{inc}$ | $maxinc$ | $mint$ | $\bar{t}$ | $maxt$ | $mink$ | $\bar{k}$ | $maxk$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 0.50 | 0.01 | 29.38 | 34.71 | 44.04 | 86.32 | 153.891 | 275.15 | 725 | 1049.20 | 1840 |
| | | | | | | 8.72 | 11.47 | 15.63 | 83 | 101.40 | 125 |
| 200 | 0.50 | 0.01 | 33.60 | 36.35 | 39.24 | 76.12 | 130.922 | 193.74 | 300 | 502.20 | 787 |
| | | | | | | 16.78 | 18.62 | 20.52 | 89 | 99.80 | 111 |
| 500 | 0.25 | 0.01 | 13.86 | 14.89 | 16.11 | 121.52 | 160.992 | 235.49 | 162 | 220.10 | 385 |
| | | | | | | 15.42 | 18.28 | 22.45 | 37 | 41.00 | 49 |
| 1000 | 0.20 | 0.01 | 10.67 | 11.13 | 11.71 | 765.34 | 1909.32 | 3284.68 | 419 | 872.20 | 1390 |
| | | | | | | 26.97 | 34.04 | 43.86 | 27 | 30.20 | 35 |

Table 3 PA1 and $\epsilon.\epsilon/2$-PA2 algorthms. $\epsilon = \delta v(P(0))$

In Table 4 we present the results by using the $\epsilon, \epsilon/2$-PA2 algorithm.

| $n$ | $\lambda$ | $\delta$ | $mininc$ | $\overline{inc}$ | $maxinc$ | $mint$ | $\bar{t}$ | $maxt$ | $mink$ | $\bar{k}$ | $maxk$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 0.20 | 0.01 | 10.67 | 11.13 | 11.71 | 26.97 | 34.04 | 43.86 | 27 | 30.20 | 35 |
| | | 0.005 | | | | 67.16 | 88.74 | 109.19 | 57 | 62.00 | 69 |
| | | 0.001 | | | | 496.35 | 673.71 | 738.98 | 297 | 312.40 | 333 |
| 1000 | 0.50 | 0.001 | 34.59 | 35.48 | 36.28 | 1738.06 | 2182.40 | 2352.48 | 1007 | 1022.20 | 1043 |
| 1500 | 0.25 | 0.001 | 14.50 | 14.77 | 14.98 | 1560.45 | 1656.31 | 1800.78 | 407 | 415.80 | 421 |

Table 4 $\epsilon, \epsilon/2$-PA2 algorithm.

In Table 5 we present the results by using the $\epsilon, \epsilon/2$-PA2 algorithm. $\epsilon = \delta v(P(0))$. Only one problem for each $n$ is presented in this table.

| $n$ | $\lambda$ | $\delta$ | $inc$ | $t$ | $k$ |
|---|---|---|---|---|---|
| 1500 | 0.50 | 0.001 | 36.97 | 4177.65 | 1059 |
| 2000 | | 0.01 | 35.92 | 628.5 | 101 |
| 2000 | | 0.005 | 36.23 | 1234.04 | 209 |
| 2500 | | 0.01 | 36.47 | 2298.41 | 109 |
| 2750 | | | 36.18 | 1613.65 | 105 |
| 3000 | | | 36.49 | 2523.59 | 107 |

Table 5 $\epsilon, \epsilon/2$-PA2 algorithm. $\epsilon = \delta v(P(0))$. Only one problem for each $n$.

#### 4.2.2 The Maximun cardinality KPB problem: $c_j = 1 \ \forall j$

Now we consider the special case in which we want to maximize the number of items selected. Thus, in this case $c_j = 1 \ \forall j$.

In Table 6 we present the results analogous to Table 3. For each $n$ the first row corresponds to the PA1 algorithm and the second row corresponds to the $\epsilon, \epsilon/2$-PA2 algorithm. Again $\epsilon = \delta v(P(0))$.

| $n$ | $\lambda$ | $\delta$ | $mininc$ | $\overline{inc}$ | $maxinc$ | $mint$ | $\bar{t}$ | $maxt$ | $mink$ | $\bar{k}$ | $maxk$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 0.50 | 0.01 | 16.46 | 18.90 | 21.08 | 86.32 | 153.891 | 275.15 | 725 | 1049.20 | 1840 |
| | | | | | | 3.42 | 4.32 | 5.21 | 41 | 46.60 | 53 |
| 200 | 0.50 | 0.01 | 15.19 | 17.34 | 20.61 | 76.12 | 130.922 | 193.74 | 300 | 502.20 | 787 |
| | | | | | | 5.43 | 6.38 | 7.60 | 41 | 44.20 | 51 |
| 500 | 0.50 | 0.01 | 16.53 | 17.47 | 18.63 | 130.40 | 162.418 | 213.15 | 218 | 267.70 | 368 |
| | | | | | | 17.61 | 19.80 | 26.38 | 39 | 45.00 | 53 |
| 1000 | 0.2000 | 0.01 | 10.67 | 11.13 | 11.71 | 765.34 | 1909.32 | 3284.68 | 419 | 872.20 | 1390 |
| | | | | | | 26.97 | 34.04 | 43.86 | 27 | 30.20 | 35 |

Table 6 PA1 and $\epsilon, \epsilon/2$-PA2 algorithms. $\epsilon = \delta v(P(0))$

Table 7 is analogous to table 4.

| $n$ | $\lambda$ | $\delta$ | $mininc$ | $\overline{inc}$ | $maxinc$ | $mint$ | $\bar{t}$ | $maxt$ | $mink$ | $\bar{k}$ | $maxk$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 0.50 | 0.01 | 16.23 | 17.17 | 17.72 | 71.66 | 96.54 | 134.63 | 43 | 47.20 | 51 |
| | | 0.005 | 16.57 | 17.52 | 18.63 | 133.58 | 194.63 | 238.04 | 85 | 91.60 | 99 |
| | | 0.001 | 16.71 | 17.76 | 18.5 | 1085.91 | 1207.58 | 1454.69 | 413 | 442.67 | 475 |
| 1500 | 0.50 | 0.001 | 16.87 | 17.73 | 18.40 | 1810.51 | 1923.16 | 2000.73 | 427 | 441.40 | 455 |
| 2000 | 0.50 | 0.005 | 16.96 | 17.41 | 18.12 | 605.31 | 648.29 | 725.62 | 89 | 95.40 | 99 |
| | | 0.001 | 17.58 | 17.92 | 18.30 | 3207.21 | 3535.88 | 4080.10 | 431 | 448.20 | 467 |
| 3000 | 0.50 | 0.005 | 17.15 | 17.54 | 17.81 | 1525.08 | 1771.49 | 2157.04 | 95 | 99.00 | 103 |
| | | 0.0025 | 17.09 | 17.41 | 17.71 | 3172.70 | 4004.68 | 4503.58 | 179 | 188.60 | 195 |

Table 7 $\epsilon, \epsilon/2$-PA2 algorithm.$\epsilon = \delta v(P(0))$

In Table 8 we present results by using the $\epsilon, \epsilon/2$-PA2 algorithm with time limit equal to 3600 seconds. We use $\delta$ to define $\epsilon = \delta v(P(0))$ and $\hat{\delta}$ was the reached tolerance when the algorithm was stopped ($\hat{\delta} v(P(0)) = max\{s_j(z^{j+1} - c^t x^j) : \ j = 1, \cdots, k\}$).

| $n$ | $\lambda$ | $\delta$ | $\hat{\delta}$ | $inc$ | k |
|---|---|---|---|---|---|
| 3000 | 0.25 | 0.0025 | 0.0025 | 7.48 | 81 |
| | 0.50 | | 0.0026 | 17.50 | 181 |
| 3500 | 0.25 | | 0.0025 | 7.25 | 79 |
| | 0.50 | | 0.0034 | 17.90 | 151 |
| 4000 | 0.25 | | 0.0025 | 7.38 | 81 |
| | 0.50 | | 0.0036 | 17.17 | 133 |

Table 8 $\epsilon, \epsilon/2$-PA2 algorithm with time limit equal to 3600 seconds.$\epsilon = \delta v(P(0))$.

# 5 Conclusions and extensions.

## 5.1 Perfomance of the algorithms

The topic that we are studying is a new one and we do not have previous experience. Our purpose was only to show that a mathematical programming approach may be useful to deal with the global reconfiguration of a system.

Our experimental results are preliminary since more problems should be solved before concluding on certain topics. We consider two problems well structured selected arbitrarily. The results are obviously biased due to the selection of the problems.

According with our results our algorithms can generate an approximation of the pareto set of the biobjective problem suggested in [1] and [2] with a reasonable computational effort. Remember that a global reconfiguration is being considered by the decision maker. We do not need a solution in seconds. We can wait a reasonable time to redefine the system. In our experiments the approximation was generated in less than an hour and a half for KPm problems with up to 1500 items and 20 constraints and for BKP problems with up to 4000 items type with the number of items of each type between 6 and 10. Note that the range of the modifications considered was very large in many cases with up to 50% and we used a very small tolerances for some problems.

The $\epsilon, \epsilon/2$-PA2 and $\epsilon, \epsilon_1$-PA3 algorithms outperformed the PA1 algorithm, by using the absolute (relative) tolerance, by far as we expected.

## 5.2 Theoretical contributions and extensions.

We designed and implemented an exact and $\epsilon$-optimal algorithms to solve the parametric problem relative to the right hand side vector of a 0-1-MILP problem when the problem has a special structure: the objective function depends only on binary variables. These algorithms may be used to redefine the system global configuration of resource constraints of 0-1-ILP problems. A standard linearization procedure was used to deal with the nonlinear terms that appear in the biobjective problem suggested in [1] and [2]. Thus, a mathematical programming approach may be used to redefine the system global configuration of resource constraints.

Our approach can be considered as an extension of the mathematical programming procedures that appear when the average shadow price (ASP) is being used to detect bottlenecks restricted to the right hand side vector. Now we include global modifications.

Let $h \in \Re^m$ with $h \geq 0$. If $\theta \in \Re$, $q^j = 0 \; \forall j$, $Q^{ij} = 0 \; \forall (i,j)$ and $d^i = h_i$ for all $i$, then $P(\theta)$ is exactly the parametric problem considered in [3]. Let $p\theta$ the price of the modification with $p \in \Re$ and $p \geq 0$. Under the same assumptions let $e(p)$ the net profit function (*npf*) defined in [10] as follows: $e(p) = max\{v(P(\theta)) - v(P(0)) - p\theta : \theta \geq 0\}$ then the *npf* allow us to know a specific nondominated solution of $BI$.

In the general case presented in this paper (with $\theta \in \Re^r$, $\underline{\theta} \leq \theta \leq \overline{\theta}$ and $p \in \Re^r$) then the *npf* may be defined analogously as $e(p) = max\{v(P(\theta)) - v(P(0)) - p^t\theta : \underline{\theta} \leq \theta \leq \overline{\theta}\}$ and again the *npf* allow us to know a specific nondominated solution of $BI$.

In [1] and [2] the authors stated:

“ Limitation 1 ASP is only applicable if objective and constraints are linear.

Limitation 2 ASP does not evaluate changes in the coefficients matrix (the matrix A) and it is only limited to RHS.

Limitation 3 ASP does not provide information about the strategy for investment in resources, and the decision maker has to manually conduct analyses to find the best investment strategy.”

We agree with Limitation 1. Now must be clear that a mathematical programming approach that is an extension of the ASP procedures may be used to overcome the limitations 2 and 3 for 0-1-ILP problems at least from the theoretical point of view and for problems with moderate size.

As we stated before our results may be extended to consider 0-1-MILP problems by using either multiobjective 0-1-MIBLP or multiobjective 0-1-MILP according to the case.

If some changes to the initial configuration must be discrete, for example with integer resource requirements ([13]) and even if some changes in the coefficients matrix must be discrete the algorithms may de redesigned without problems by using $\theta$ appropriately.

Finally, the price of the modification may be defined as a piecewise linear function by using standard methods to define the linearization of the problems.

# References

[1] Mohammad Reza Bonyadi and Zbigniew Michalewicz: Evolutionary Computation for Real-World Problems *Challenges in Computational Statistics and Data Mining. Stan Matwin-Jan Mielnikzuk Editors, Studies in Computational Intelligence 605, Springer Verlag* pp 1-21, 2014.

[2] Mohammad Reza Bonyadi, Zbigniew Michalewicz and Markus Wagner: Beyond the Edge of Feasibility: Analysis of Bottlenecks. *Simulated Evolution and Learning* Volume 8886 of the series Lecture Notes in Computer Science pp 431-442, 2014.

[3] Alejandro Crema: Average shadow price in a mixed integer linear programming problem. *European Journal of Operational Research* Volume 85, Issue 3, 21 September 1995, Pages 625-635.

[4] Alejandro Crema: A procedure to verify the completeness of the right-hand-side parametric analysis for a mixed integer linear programming problem

*European Journal of Operational Research* Volume 108, Issue 3, 1 August 1998, Pages 684-695.

[5] Alejandro Crema: An algorithm to perform a complete right-hand-side parametrical analysis for a 0-1-integer linear programming problem. *European Journal of Operational Research* Volume 114, Issue 3, 1 May 1999, Pages 569579.

[6] Arnaud Fréville: The multidimensional 01 knapsack problem: An overview. *European Journal of Operational Research* Volume 155, Issue 1, 16 May 2004, Pages 121.

[7] Fred Glover: Improved linear integer programming formulations of nonlinear integer problems *Management Science* Vol. 22, No. 4, December, pp. 455-460, 1975

[8] Akshay Gupte, Shabbir Ahmed, Myun Seok Cheon and Santanu Dey: Solving mixed integer bilinear problems using milp formulations. *SIAM J. Optim., 23(2)*, pp 721744, 2013

[9] Sehun Kim and Seong-cheol Cho: A shadow price in integer programming for management decision. *European Journal of Operational Research* Volume 37, Issue 3, December 1988, Pages 328-335.

[10] Sehun Kim and Seong-cheol Cho: Average shadow prices in mathematical programming. *Journal of Optimization Theory and Applications* July 1992, Volume 74, Issue 1, pp 57-74.

[11] Silvano Martello and Paolo Toth: Knapsack Problems, Algorithms and computer implementations *John Wiley & Sons,* 1990.

[12] Saral Mukherjee and A.K. Chatterjeeb: Unified Concept of Bottleneck. *Indian Institute of Management Ahmedabad 380015,INDIA* W.P. No. 2006-05-01, May 2006.

[13] Saral Mukherjee and A.K. Chatterjeeb: The average shadow price for MILPs with integral resource availability and its relationship to the marginal unit shadow price. *European Journal of Operational Research* Volume 169, Issue 1, 16 February 2006, Pages 5364.

[14] Alexander Mitsos and Paul I. Barton: Parametric mixed-integer 0-1 linear programming: The general case for a single parameter. *European Journal of Operational Research* Volume 108, Issue 3, 1 August 1998, Pages 684-695.

[15] Jakob Puchinger and Günther Raidl, Ulrich Pferschy: The Multidimensional Knapsack Problem: Structure and Algorithms. *INFORMS Journal on Computing, Institute for Operations Research and the Management Sciences (INFORMS)*, 2010, 22 (2), pp.250 - 265.

[16] John Sylva and Alejandro Crema: A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs.*European Journal of Operational Research* Volume 180, Issue 3, 1 August 2007, Pages 10111027.

[17] Estefania Yap: A literature review of multi-objective programming. *Australian Mathematical Sciences Institute* Available via WEB: Article (2010).