DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

A new class of functions for measuring solution integrality in the Feasibility Pump approach

Marianna De Santis
Stefano Lucidi
Francesco Rinaldi

Technical Report n. 8, 2011

A new class of functions for measuring solution integrality in the Feasibility Pump approach

M. De Santis, S. Lucidi, F. Rinaldi

Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma
Via Ariosto, 25 - 00185 Roma - Italy

e-mail: mdesantis@dis.uniroma1.it
e-mail: stefano.lucidi@dis.uniroma1.it
e-mail: rinaldi@dis.uniroma1.it

Abstract

Mixed-Integer optimization is a powerful tool for modeling many optimization problems arising from real-world applications. Finding a first feasible solution represents the first step for several MIP solvers. The Feasibility pump is a heuristic for finding feasible solutions to mixed integer linear problems which is effective even when dealing with hard MIP instances. In this work, we start by interpreting the Feasibility Pump as a Frank-Wolfe method applied to a nonsmooth concave merit function. Then, we define a general class of functions that can be included in the Feasibility Pump scheme for measuring solution integrality and we identify some merit functions belonging to this class. We further extend our approach by dynamically combining two different merit functions. Finally, we define a new version of the Feasibility Pump algorithm, which includes the original version of the Feasibility Pump as a special case, and we present computational results on binary MILP problems showing the effectiveness of our approach.

Keywords. Mixed integer programming, Concave penalty functions, Frank-Wolfe algorithm, Feasibility Problem.

MSC. 90C06, 90C10, 90C11, 90C30, 90C59

1 Introduction

Many real-world problems can be modeled as Mixed Integer Programming (MIP) problems, namely as minimization problems where some (or all) of the variables only assume integer values. Finding quickly a first feasible solution is crucial for solving this class of problems. In fact, many local-search approaches for MIP problems such as Local Branching [16], guide dives and RINS [12] can be used only if a feasible solution is available.

In the literature, several heuristics methods for finding a first feasible solution for a MIP problem have been proposed (see e.g. [3]-[5], [8], [18]-[21], [23], [26]). Recently, Fischetti, Glover and Lodi [15] proposed a new heuristic, the well-known Feasibility Pump, that turned out to be very useful in finding a first feasible solution even when dealing with hard MIP instances. The FP heuristic is implemented in various MIP solvers such as BONMIN [9].

The basic idea of the FP is that of generating two sequences of points $\{\bar{x}^k\}$ and $\{\tilde{x}^k\}$ such that $\bar{x}^k$ is LP-feasible, but may not be integer feasible, and $\tilde{x}^k$ is integer, but not necessarily LP-feasible. To be more specific the algorithm starts with a solution of the LP relaxation $\bar{x}^0$ and sets $\tilde{x}^0$ equal to the rounding of $\bar{x}^0$. Then, at each iteration $\bar{x}^{k+1}$ is chosen as the nearest LP-feasible point in ℓ_1-norm to $\tilde{x}^k$, and $\tilde{x}^{k+1}$ is obtained as the rounding of $\bar{x}^{k+1}$. The aim of the algorithm is to reduce at each iteration the distance between the points of the two sequences, until the two points are the same and an integer feasible solution is found. Unfortunately, it can happen that the distance between $\bar{x}^{k+1}$ and $\tilde{x}^k$ is greater than zero and $\tilde{x}^{k+1} = \tilde{x}^k$, and the strategy can stall. In order to overcome this drawback, random perturbations and restart procedures are performed.

As the algorithm has proved to be effective in practice, various papers devoted to its further improvements have been developed. Fischetti, Bertacco and Lodi [7] extended the ideas on which the FP is based in two different directions: handling MIP problems with both 0-1 and integer variables, and exploiting the FP information to drive a subsequent enumeration phase. In [1], in order to improve the quality of the feasible solution found, Achterberg and Berthold consider an alternative distance function which takes into account the original objective function. In [17], Fischetti and Salvagnin proposed a new rounding heuristic based on a diving-like procedure and constraint propagation. The Feasibility Pump has been further extended to the case of mixed integer nonlinear programming problems in [10, 11].

In [8], J.Eckstein and M.Nediak noticed that the FP heuristic may be seen as a form of Frank-Wolfe procedure applied to a nonsmooth merit function which penalizes the violation of the 0-1 constraints. In practice, the Feasibility Pump combines a local algorithm (namely the Frank-Wolfe algorithm) with a suitably developed perturbing procedure for solving a specific global optimization problem:

$$x^* = \arg\min\{f(x) : \, x \in P\},$$

where P is the relaxation of the feasible set of the original MIP Problem and $f(x)$ is a function penalizing the violation of the integrality constraints. Therefore the Feasibility Pump can be seen as a form of Iterated Local Search or Basin Hopping algorithm (see e.g. [6, 25, 27]).

In this paper, we analyze in deep the relationship between the Feasibility Pump and the Frank-Wolfe algorithm. In this context, we define a new class of merit functions that can be included in the basic FP scheme [15]. A reported extended computational experience seems to indicate that the use of these new merit functions improves the FP efficiency.

The paper is organized as follows. In Section 2, we give a brief review of the Feasibility Pump heuristic. In Section 3, we show the equivalence between the FP heuristic and the Frank-Wolfe algorithm applied to a nonsmooth merit function. In Section 4, we define a new class of merit functions for measuring the solution integrality, we introduce new nonsmooth merit functions and we discuss their properties. We present our algorithm in Section 5. In Section 6, we extend our approach by dynamically combining two different merit functions. Computational results are shown in Section 7, where we give a detailed performance comparison of our algorithm with the FP. Further, we show that using somehow more than one merit function at time can improve the efficiency of the algorithm. Some conclusions are drawn in Section 8.

In the following, given a concave function $f : R^n \to R$, we denote by $\partial f(x)$ the set of supergradients of f at the point x, namely

$$\partial f(x) = \{v \in R^n \ : \ f(y) - f(x) \leq v^T(y - x), \ \forall \ y \in R^n\}.$$

2 The Feasibility Pump Heuristic

We consider a MIP problem of the form:

$$\begin{aligned} \min\, & c^T x \\ \text{s.t. } & Ax \geq b \\ & x_j \in \{0,1\} \ \forall j \in I, \end{aligned} \tag{MIP}$$

where $A \in \mathbf{R}^{m\times n}$ and $I \subset \{1, 2, \ldots, n\}$ is the set of indices of zero-one variables. Let $P = \{x : Ax \geq b,\, 0 \leq x_j \leq 1, \forall\, j \in I\}$ denote the polyhedron of the LP-relaxation of (MIP). The Feasibility Pump starts from the solution of the LP relaxation problem $\bar{x}^0 := \arg\min\{c^T x : x \in P\}$ and generates two sequences of points $\bar{x}^k$ and $\tilde{x}^k$: $\bar{x}^k$ is LP-feasible, but may be integer infeasible; $\tilde{x}^k$ is integer, but not necessarily LP-feasible. At each iteration $\bar{x}^{k+1} \in P$ is the nearest point in ℓ_1-norm to $\tilde{x}^k$:

$$\bar{x}^{k+1} := \arg\min_{x\in P} \, \Delta(x, \tilde{x}^k)$$

where

$$\Delta(x, \tilde{x}^k) = \sum_{j\in I} |x_j - \tilde{x}^k_j|.$$

The point $\tilde{x}^{k+1}$ is obtained as the rounding of $\bar{x}^{k+1}$. The procedure stops if at some index l, $\bar{x}^l$ is integer or, in case of failing, if it reaches a time or iteration limit. In order to avoid stalling issues and loops, the Feasibility Pump performs a perturbation step. Here we report a brief outline of the basic scheme:

The Feasibility Pump (FP) - basic version

Initialization: Set $k = 0$, let $\bar{x}^0 := \arg\min\{c^T x : x \in P\}$

While (not stopping condition) **do**

> **Step 1 If** ($\bar{x}^k$ is integer) return $\bar{x}^k$
>
> **Step 2** Compute $\tilde{x}^k = round(\bar{x}^k)$
>
> **Step 3 If** (cycle detected) $perturb(\tilde{x}^k)$
>
> **Step 4** Compute $\bar{x}^{k+1} := \arg\min\{\Delta(x, \tilde{x}^k) : x \in P\}$
>
> **Step 5** Update $k = k + 1$

End While

Now we give a better description of the rounding and the perturbing procedures used respectively at **Step 2** and at **Step 3** (See e.g. [7], [15]):

Round: This function transforms a given point $\bar{x}^k$ into an integer one, $\tilde{x}^k$. The easiest choice is that of rounding each component $\bar{x}^k_j$ with $j \in I$ to the nearest integer, while leaving the continuous components of the solution unchanged. Formally,

$$\tilde{x}^k_j = \begin{cases} [\bar{x}^k_j] & \text{if } j \in I \\ \\ \bar{x}^k_j & \text{otherwise} \end{cases} \tag{1}$$

where $[\cdot]$ represents scalar rounding to the nearest integer.

Perturb: The aim of the perturbation procedure is to avoid cycling and it consists in two heuristics. To be more specific:

- if $\tilde{x}^k_j = \tilde{x}^{k+1}_j$ for all $j \in I$ a weak perturbation is performed, namely, a random number of integer constrained components, chosen as to minimize the increase in the distance $\Delta(\bar{x}^{k+1}, \tilde{x}^{k+1})$, is flipped.
- If a cycle is detected by comparing the solutions obtained in the last 3 iterations, or in any case after R iterations, a strong random perturbation is performed. For each $j \in I$ a uniformly random value is generated, $\rho_j \in [-0.3, 0.7]$ and if

  $$|\bar{x}^{k+1}_j - \tilde{x}^{k+1}_j| + \max\{\rho_j, 0\} > 0.5$$

  the component $\tilde{x}^{k+1}_j$ is flipped.

Remark 1 *The objective function $\Delta(x, \tilde{x}^k)$ discourages the optimal solution of the relaxation from being "too far" from $\tilde{x}^k$. In practice, the method tries to force a large number of variables of $\bar{x}^{k+1}$ to have the same (integer) value as $\tilde{x}^k$ (see [15]).*

3 The FP heuristic as a Frank-Wolfe algorithm for minimizing a nonsmooth merit function

In a recent work J.Eckstein and M.Nediak [8] noticed that the feasibility pump heuristic may be seen as a nonsmooth Frank-Wolfe merit function procedure. In order to better understand this equivalence we recall the unitary stepsize Frank-Wolfe method for concave non-differentiable functions. Let us consider the problem

$$\begin{aligned} &\min f(x) \\ &x \in P \end{aligned} \tag{2}$$

where $P \subset R^n$ is a non empty polyhedral set that does not contain lines going to infinity in both directions, $f : R^n \to R$ is a concave, non-differentiable function, bounded below on P. The Frank-Wolfe algorithm with unitary stepsize can be described as follows.

Frank-Wolfe - Unitary Stepsize (FW1) Algorithm

Initialization: Set $k = 0$, let $x^0 \in R^n$ be the starting point, compute $g^0 \in \partial f(x^0)$
While $x^k \notin \arg\min_{x \in P} (g^k)^T x$

Step 1 Compute a vertex solution x^{k+1} of

$$\min_{x \in P} (g^k)^T x$$

Step 2 Compute $g^{k+1} \in \partial f(x^{k+1})$, update $k = k + 1$

End While

The algorithm involves only the solution of linear programming problems, and the following result, proved in [29], shows that the algorithm generates a finite sequence and that it terminates at a stationary point $x^\star$, namely a point satisfying the following condition:

$$(g^\star)^T (x - x^\star) \geq 0, \;\; \forall x \in P \tag{3}$$

with $g^\star \in \partial f(x^\star)$.

Proposition 1 *The Frank-Wolfe algorithm with unitary stepsize converges to a vertex stationary point of problem* (2) *in a finite number of iterations.*

Now we consider the basic FP heuristic without any perturbation (i.e. without Step 3) and we show that it can be interpreted as the Frank-Wolfe algorithm with unitary stepsize applied to a concave, nondifferentiable merit function.
First of all, we can easily see that

$$\Delta(x, \tilde{x}^k) = \sum_{j \in I : \tilde{x}_j^k = 0} x_j - \sum_{j \in I : \tilde{x}_j^k = 1} x_j.$$

At each iteration, the Feasibility Pump for mixed 0-1 problems computes, at Step 2, the rounding of the solution $\bar{x}^k$, thus giving $\tilde{x}^k$. Then, at Step 4, it computes the solution of the LP problem

$$\begin{aligned} \bar{x}^{k+1} \in \arg\min & \, \Delta(x, \tilde{x}^k) \\ \text{s.t.} & \;\; Ax \geq b \\ & \;\; 0 \leq x_j \leq 1 \; \forall j \in I. \end{aligned} \tag{4}$$

These two operations can be included in the unique step of calculating the solution of the following LP problem:

$$\begin{aligned} \min & \sum_{j \in I : \bar{x}_j^k < \frac{1}{2}} x_j - \sum_{j \in I : \bar{x}_j^k \geq \frac{1}{2}} x_j \\ \text{s.t.} & \;\; Ax \geq b \\ & \;\; 0 \leq x_j \leq 1 \; \forall j \in I. \end{aligned} \tag{5}$$

Since the function

$$v(t) = \begin{cases} 1 & \text{if } t < \frac{1}{2} \\ -1 & \text{if } t \geq \frac{1}{2} \end{cases} \tag{6}$$

is such that $v(t) \in \partial \min\{t, 1-t\}$, Problem (5) can be seen as a generic iteration of the Frank Wolfe method with unitary stepsize applied to the following minimization problem

$$\begin{aligned} \min \ & \sum_{i \in I} \min\{x_i, 1 - x_i\} \\ \text{s.t.} \ & Ax \geq b \\ & 0 \leq x_i \leq 1 \ \forall i \in I. \end{aligned} \tag{7}$$

4 New nonsmooth merit functions for the FP approach

As we have seen in the previous section, the basic Feasibility Pump is equivalent to minimizing a separable nonsmooth function which penalizes the 0-1 infeasibility, namely

$$f(x) = \sum_{i \in I} \min\{x_i, 1 - x_i\}. \tag{8}$$

When using the Frank Wolfe unitary stepsize algorithm for solving Problem (7), at each iteration, if x^k is not a stationary point, we get a new point x^{k+1} such that

$$(g^k)^T (x^{k+1} - x^k) < 0,$$

with $g^k \in \partial f(x^k)$. Then, from the concavity of the objective function we have

$$f(x^{k+1}) \leq f(x^k) + (g^k)^T (x^{k+1} - x^k) < f(x^k), \tag{9}$$

which means that at each iteration a reduction of the merit function is obtained. Anyway, this might not correspond to a reduction in the number of variables that violate integrality.

Example 1 *Let us consider the following two points*

$$x = \left(0, \frac{1}{2}, 0, 0\right)^T; \qquad y = \left(0, \frac{1}{6}, \frac{1}{6}, 0\right)^T.$$

Let f be the function defined in (8). It is easy to notice that

$$f(y) < f(x),$$

but the number of noninteger components of y is greater than the number of noninteger components of x.

As the main goal is finding an integer feasible solution, it would be better to use a function having the following features:

(i) it decreases whenever the number of integer variables increases;

(ii) if it decreases, then the number of noninteger variables does not increase.

A function satisfying these features is the following:

$$\psi(x) = card\{\, x_i : i \in I,\ x_i \notin \{0, 1\}\,\}. \tag{10}$$

The function (10) can be rewritten as:

$$\psi(x) = \sum_{i \in I} s(\min\{x_i, 1 - x_i\}) \tag{11}$$

where $s : \mathbf{R} \rightarrow \mathbf{R}^+$ is the *step function*:

$$s(t) = \begin{cases} 1 & \text{if } t > 0 \\ \\ 0 & \text{otherwise.} \end{cases}$$

Since the step function is a nonconvex and discontinuous function, minimizing (11) over a polyhedral set is a very hard problem. In the following we prove a general result to define approximations of function (11) that are easier to handle from a computational point of view and guarantee satisfaction of (i) and (ii) when evaluated on the vertices of a polyhedron.

Proposition 2 *Let $V \subset [0,1]^n$ be the set of vertices of a polytope $P = \{ x : Ax \geq b, x \in [0,1] \}$. Let α_l and α_u be the following values:*

$$\alpha_l = \min_{x \in V} l(x)$$

$$\alpha_u = \min_{x \in V} u(x)$$

where

$$l(x) = \begin{cases} \min\{ x_i : i = 1, \dots, n; \, x_i \neq 0\} & \text{if } x \neq 0 \\ 1 & \text{if } x = 0; \end{cases}$$

$$u(x) = \begin{cases} \max\{ x_i : i = 1, \dots, n; \, x_i \neq 1\} & \text{if } x \neq e \\ 1 & \text{if } x = e. \end{cases}$$

Let $\phi : [0,1]^n \rightarrow \mathbf{R}$ be a separable function

$$\phi(x) = \sum_{i \in I} \varphi(x_i). \tag{12}$$

We assume that $\varphi : [0,1] \rightarrow R$ satisfies the following:

1)

$$\varphi(0) = \varphi(1); \tag{13}$$

2) there exists an $M > 0$ such that

(i) for $\bar{\alpha} \in \{0,1\}$ and $\tilde{\alpha} \in [\alpha_l, \alpha_u]$ we have

$$\varphi(\bar{\alpha}) - \varphi(\tilde{\alpha}) \leq -M; \tag{14}$$

(ii) for $\bar{\alpha}, \tilde{\alpha} \in [\alpha_l, \alpha_u]$ we have

$$|\varphi(\bar{\alpha}) - \varphi(\tilde{\alpha})| \leq \frac{M}{n}. \tag{15}$$

Then, for $x, y \in V$:

a) $\psi(x) < \psi(y)$ implies $\phi(x) < \phi(y)$;

b) $\phi(x) < \phi(y)$ implies $\psi(x) \leq \psi(y)$.

Proof.

a) We consider two points $x, y \in V$ such that $\psi(x) < \psi(y)$. We can define two sets of indices related to the non-integer components of x and y:

$$U = \{i \in \{1, \dots, n\} \mid i \in I,\, x_i \notin \{0,1\}\},$$
$$W = \{j \in \{1, \dots, n\} \mid j \in I,\, y_j \notin \{0,1\}\}.$$

Then we can write

$$\begin{aligned}\phi(x) - \phi(y) &= \sum_{i\in I} \varphi(x_i) - \sum_{j\in I} \varphi(y_j) = \\ &= \sum_{i\in U} \varphi(x_i) + \sum_{i\in I\setminus U} \varphi(x_i) - \sum_{j\in W} \varphi(y_j) - \sum_{j\in I\setminus W} \varphi(y_j).\end{aligned} \tag{16}$$

Since $\psi(x) < \psi(y)$, we have that

$$|U| < |W|$$

and

$$|I \setminus U| > |I \setminus W|.$$

Let us assume, without loss of generality, that

$$|W| - |U| = 1,$$

and there exists an index $\bar{\jmath}$ such that

$$W \setminus \{\bar{\jmath}\} = U$$
$$(I \setminus U) \setminus \{\bar{\jmath}\} = I \setminus W.$$

Then we can write

$$\begin{aligned}\phi(x) - \phi(y) = \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) &+ \sum_{j\in U} \varphi(x_j) + \sum_{\substack{j\in I\setminus U\\ j\neq\bar{\jmath}}} \varphi(x_j) - \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} \varphi(y_j) - \sum_{j\in I\setminus W} \varphi(y_j) = \\ = \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) &+ \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} \varphi(x_j) + \sum_{\substack{j\in I\setminus U\\ j\neq\bar{\jmath}}} \varphi(x_j) - \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} \varphi(y_j) - \sum_{\substack{j\in I\setminus U\\ j\neq\bar{\jmath}}} \varphi(y_j) = \\ = \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) &+ \sum_{\substack{j\in I\setminus U\\ j\neq\bar{\jmath}}} (\varphi(x_j) - \varphi(y_j)) + \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} (\varphi(x_j) - \varphi(y_j)) \leq \\ \leq \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) &+ \sum_{\substack{j\in I\setminus U\\ j\neq\bar{\jmath}}} (\varphi(x_j) - \varphi(y_j)) + \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} |\varphi(x_j) - \varphi(y_j)|\end{aligned} \tag{17}$$

By using (13) we obtain

$$\phi(x) - \phi(y) \leq \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) + \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} |\varphi(x_j) - \varphi(y_j)|. \tag{18}$$

Now we notice that $x_{\bar{\jmath}} \in \{0, 1\}$, $y_{\bar{\jmath}} \in [\alpha_l, \alpha_u]$ and $x_j, y_j \in [\alpha_l, \alpha_u]$ for all $j \in W \setminus \{\bar{\jmath}\}$. Then, by using (14) and (15), we have

$$\phi(x) - \phi(y) \leq \varphi(x_{\bar{\jmath}}) - \varphi(y_{\bar{\jmath}}) + \sum_{\substack{j\in W\\ j\neq\bar{\jmath}}} |\varphi(x_j) - \varphi(y_j)| \leq -M + (|I| - 1)\frac{M}{n} < 0. \tag{19}$$

Hence we have

$$\phi(x) < \phi(y).$$

b) We assume by contradiction that there exist two points $x, y \in V$ such that $\phi(x) < \phi(y)$ and

$$\psi(x) > \psi(y). \tag{20}$$

By (20), recalling the first part of the proof, we have that $\phi(x) > \phi(y)$, which contradicts our initial assumption.

□

Summarizing, if an approximation $\phi(x)$ satisfying the assumptions of Proposition 2 is available, we can solve, in place of the original FP problem (7), the following problem

$$\begin{aligned} \min\ & \phi(x) = \sum_{i \in I} \varphi(x_i) \\ \text{s.t.}\ & Ax \geq b \\ & 0 \leq x_i \leq 1\ \forall i \in I. \end{aligned} \tag{21}$$

As the method we use for solving the minimization problem stated above is the Frank-Wolfe algorithm, which at each step moves from a vertex to another guaranteeing the reduction of the chosen approximation, we have (by point b) of Proposition 2) that, at each iteration of the algorithm, the number of the noninteger components of the current solution does not increase. Taking into account Proposition 2 and the ideas developed in [28, 32], we consider the following $\varphi(\cdot)$ terms to be used in the objective function of problem (21):

Logarithmic function

$$\varphi(t) = \min\big\{ \ln(t+\varepsilon), \ln[(1-t)+\varepsilon]\big\} \tag{22}$$

Hyperbolic function

$$\varphi(t) = \min\big\{ -(t+\varepsilon)^{-p}, -[(1-t)+\varepsilon]^{-p}\big\} \tag{23}$$

Exponential function

$$\varphi(t) = \min\big\{1 - \exp(-\alpha t), 1 - \exp(-\alpha(1-t))\big\} \tag{24}$$

Logistic function

$$\varphi(t) = \min\big\{[1+\exp(-\alpha t)]^{-1}, [1+\exp(-\alpha(1-t))]^{-1}\big\} \tag{25}$$

where $\varepsilon, \alpha, p > 0$. In Fig. 1, we compare the φ term related to the FP heuristic with those given by (22)-(25).

Now we prove that, for a particular choice of the φ term, the assumptions of Proposition 2 are satisfied.

Proposition 3 *For the term (22), there exists a value $\bar{\varepsilon} > 0$ such that for any $\varepsilon \in (0, \bar{\varepsilon}]$ assumptions* 1) *and* 2) *of Proposition 2 are satisfied.*

Proof. It can be easily noticed that when $x \in \{0, 1\}$ we have

$$\varphi(x) = \ln \varepsilon,$$

then assumption 1) of Proposition 2 is satisfied.

Now, without any loss of generality, we suppose

$$\alpha_l = \min\{\alpha_l, 1 - \alpha_u\} \tag{26}$$

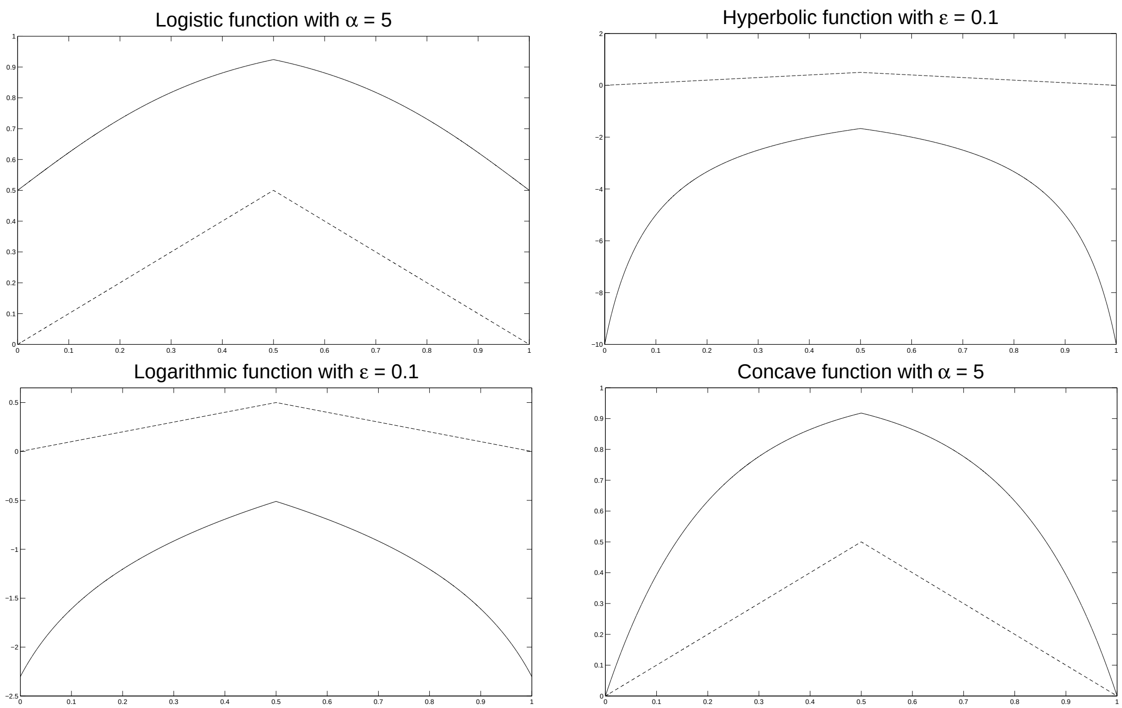


Figure 1: Comparison between the original FP term (dashed line) and the new terms (solid line).

and we notice that there exists a value $\bar{\varepsilon} > 0$ such that for any $\varepsilon \in (0, \bar{\varepsilon}]$ the following inequality holds:

$$\ln \varepsilon - \ln(\alpha_l + \varepsilon) + n(\ln(1/2 + \varepsilon) - \ln(\alpha_l + \varepsilon)) \leq 0. \tag{27}$$

As the function $\varphi(t)$ is strictly increasing in $[0, \frac{1}{2}]$ and strictly decreasing in $(\frac{1}{2}, 1]$ and it is symmetric with respect to the point $t = \frac{1}{2}$, we have for $\bar{\alpha} \in \{0, 1\}$ and $\tilde{\alpha} \in [\alpha_l, \alpha_u]$

$$\varphi(\bar{\alpha}) - \varphi(\tilde{\alpha}) \leq \varphi(0) - \varphi(\alpha_l).$$

Then we set

$$M = \varphi(\alpha_l) - \varphi(0) = \ln(\alpha_l + \varepsilon) - \ln \varepsilon, \tag{28}$$

and (i) in Assumption 2) of Proposition 2 is satisfied.

As the maximum of $\varphi(t)$ is attained at $t = \frac{1}{2}$ and due to the structure of function $\varphi(t)$, we have for any choice of $\bar{\alpha}, \tilde{\alpha} \in [\alpha_l, \alpha_u]$:

$$|\varphi(\bar{\alpha}) - \varphi(\tilde{\alpha})| \leq \varphi(1/2) - \varphi(\alpha_l). \tag{29}$$

Since $ii)$ in Assumption 2) needs to be verified for any choice of $\bar{\alpha}, \tilde{\alpha} \in [\alpha_l, \alpha_u]$, by (29) it is sufficient to show that

$$\varphi(1/2) - \varphi(\alpha_l) \leq \frac{M}{n}.$$

By using (28) and (27), we can easily verify that for any $\varepsilon \in (0, \bar{\varepsilon}]$, the following inequality holds:

$$\varphi(0) - \varphi(\alpha_l) + n(\varphi(1/2) - \varphi(\alpha_l)) = \tag{30}$$

$$= \ln \varepsilon - \ln(\alpha_l + \varepsilon) + n(\ln(1/2 + \varepsilon) - \ln(\alpha_l + \varepsilon)) \leq 0.$$

Then (ii) in Assumption 2) of Proposition 2 is satisfied. □

The result proved in Proposition 3 for the term (22) can also be proved for the terms (23)-(25) repeating the same arguments, thus all the merit functions (22)-(25) are suitable to penalize the number of variables that violate the integrality constraints.

We remark that functions (22)-(25) have also another interesting theoretical property: they can be used in an exact penalty approach like that proposed in [28]. In fact, it is possible to prove that terms (22)-(25) can be used to transform a MIP problem into an equivalent continuous problem:

Proposition 4 *Let f be a Lipschitz continuous function bounded on P. For every penalty term*

$$\phi(x) = \sum_{i \in I} \varphi(x_i)$$

with φ as in (22)-(25) a value $\bar{\varepsilon} > 0$ exists such that, for any $\varepsilon \in]0, \bar{\varepsilon}]$, problem

$$\min f(x), \quad s.t. \quad x \in P, \;\; x_i \in \{0,1\}, \;\; \forall i \in I \tag{31}$$

and problem

$$\min f(x) + \tilde{\phi}(x,\varepsilon), \quad s.t. \quad x \in P, \;\; 0 \le x_i \le 1, \;\; \forall i \in I \tag{32}$$

where

$$\tilde{\phi}(x,\varepsilon) = \begin{cases} \phi(x) & \text{if } \phi \text{ is given by (8) and } \varphi \text{ by (22)-(23)} \\ \dfrac{1}{\varepsilon}\, \phi(x) & \text{if } \phi \text{ is given by (8) and } \varphi \text{ by (24)-(25)} \end{cases}$$

have the same minimum points.

Proof. the proof follows the same arguments as in [28]. See Appendix A for further details. □

This result suggests that these new merit functions can be used to define new Feasibility Pump heuristics that improve the quality of the solution in terms of objective function value like those proposed in [1] and [8]. In fact, the heuristic proposed in [1] can be seen as a Frank-Wolfe algorithm applied to problem (32) with the penalty term (8). Furthermore, the restarting rules used in the Feasibility Pump algorithm can be reinterpreted as techniques for escaping from noninteger stationary points.
We can also include these functions into an algorithmic framework to determine the minimizer of a nonlinear programming problem with integer variables (see e.g. [31]). Anyway, the use of the continuous reformulation of the original mixed integer problem is beyond the scope of this paper and will be the subject of a future work.
In the next Section we will focus on finding a first feasible solution to a MIP problem. In particular, we tackle problem (21) by a modified Feasibility Pump approach based on the concave functions described above.

5 A reweighted version of the Feasibility Pump heuristic

The use of the merit functions (22)-(25) defined in the previous section leads to a new FP scheme in which the ℓ_1-norm used for calculating the next LP-feasible point is replaced with a "weighted" ℓ_1-norm of the form

$$\Delta_W(x,\tilde{x}) = \sum_{j \in I} w_j |x_j - \tilde{x}_j| = \|W(x - \tilde{x})\|_1, \tag{33}$$

where

$$W = diag(w_1, \dots, w_n)$$

and w_j, $j = 1, \dots, n$ are positive weights depending on the merit function ϕ chosen. The main feature of the method is the use of an infeasibility measure that

- tries to discourage the optimal solution of the relaxation from being far from $\tilde{x}$ (similarly to the original FP algorithm);
- takes into account, in some way, the information carried by the LP-feasible points obtained at the previous iterations of the algorithm for speeding up the convergence to 0-1 feasible points.

Here we report an outline of the algorithm:

Reweighted Feasibility Pump (RFP) - basic version

Initialization: Set $k = 0$, let $\bar{x}^0 := \arg\min\{c^T x : x \in P\}$

While (not stopping condition) **do**

> **Step 1 If** ($\bar{x}^k$ is integer) return $\bar{x}^k$
>
> **Step 2** Compute $\tilde{x}^k = round(\bar{x}^k)$
>
> **Step 3 If** (cycle detected) $perturb(\tilde{x}^k)$
>
> **Step 4** Compute $\bar{x}^{k+1} := \arg\min\{\|W^k(x - \tilde{x}^k)\|_1 : x \in P\}$
>
> **Step 5** Update $k = k + 1$

End While

We assume that the *round* and *perturb* procedures are the same as those described in Section 2 for the original version of the FP heuristic. Anyway, different rounding and perturbing procedures can be suitably developed.

Following the same reasoning of Section 3, we can reinterpret the reweighted FP heuristic without perturbation as the unitary stepsize Frank-Wolfe algorithm applied to the merit function ϕ. Let us now consider a generic iteration k of the reweighted FP. At Step 2, the algorithm rounds the solution $\bar{x}^k$, thus giving $\tilde{x}^k$. Then, at Step 4, it computes the solution of the LP problem

$$\begin{aligned} \bar{x}^{k+1} \in \arg\min\ & \Delta_{W^k}(x, \tilde{x}^k) \\ \text{s.t.}\ & Ax \geq b \\ & 0 \leq x_j \leq 1\ \forall j \in I. \end{aligned} \tag{34}$$

Similarly to the FP algorithm, these two operations can be included in the unique step of calculating the solution of the following LP problem:

$$\begin{aligned} \min\ & \sum_{j \in I : \bar{x}_j^k < \frac{1}{2}} w_j^k x_j - \sum_{j \in I : \bar{x}_j^k \geq \frac{1}{2}} w_j^k x_j \\ \text{s.t.}\ & Ax \geq b \\ & 0 \leq x_j \leq 1\ \forall j \in I. \end{aligned} \tag{35}$$

By setting

$$w_j^k = |g_j^k|$$

with $g^k \in \partial\phi(\bar{x}^k)$, Problem (35), as we have already said, can be seen as the iteration of the Frank Wolfe method with unitary stepsize applied to the minimization problem (21).

In order to highlight the differences between the ℓ_1-norm and the weighted ℓ_1-norm we report the following example:

Example 2 *Consider the MILP problem:*

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & x \in P \\ & x \in \{0,1\}^3 \end{aligned} \tag{36}$$

where $P \subset [0,1]^3$ is the polyhedron in Fig. 2. Let $x^L = \left(\frac{9}{20}, \frac{1}{8}, \frac{1}{8}\right)$ be the solution of the linear relaxation of (36) and $x^I = (0,0,0)$ be its rounding. The minimization of $\Delta(x, x^I) = \|x - x^I\|_1$

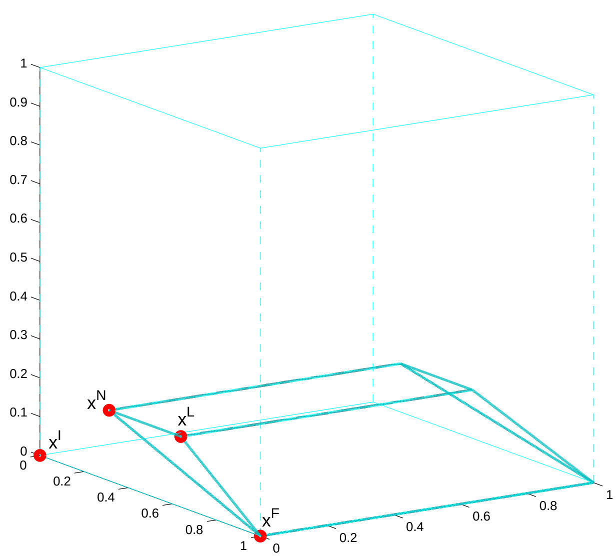


Figure 2: Feasible set of Problem 36.

over P leads to $x^N = \left(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right)$, since $\|x^N - x^I\|_1 < \|x - x^I\|_1$, for all $x \in P$.
Consider now the weighted ℓ_1-norm obtained using the logarithmic merit function

$$\phi(x) = \sum_{i \in I} \min\left\{ \ln(x_i + \varepsilon), \ln[(1 - x_i) + \varepsilon]\right\},$$

where ε is a small positive value. By minimizing the weighted distance between x and x^I over P, we obtain the point $x^F = (1,0,0)$. In fact, we have

$$\Delta_W(x^F, x^I) < \Delta_W(x, x^I),$$

for all $x \in P$. Thus the ℓ_1-norm finds a solution which does not satisfy the integrality constraints, while the reweighted ℓ_1-norm gets an integer feasible solution.

Finally, we want to remark that the original Feasibility Pump Algorithm is a special case of the Reweighted Feasibility Pump obtained by setting $W^k = I$.

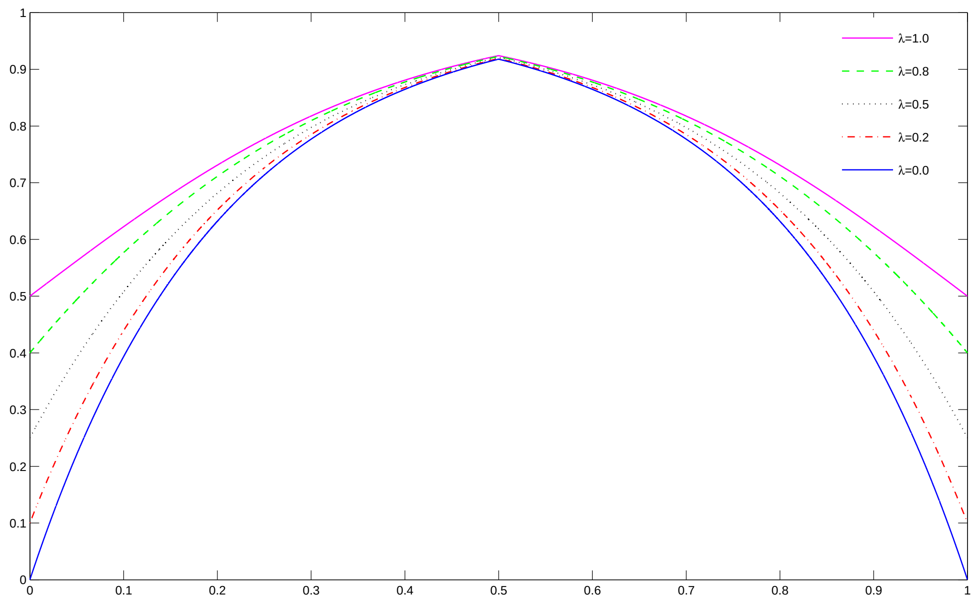


Figure 3: Behaviour of the function obtained combining exponential and logistic function

6 Combining Two Merit Functions

As we have already said, the main drawback of the FP heuristic is its tendency to stall (i.e. to get stuck in a point that is not an integer feasible solution). For this reason, a random perturbation (or a restart) is performed. A good idea might be that of modifying the objective function (in addiction to the random perturbation/restart usually adopted) any time the algorithm stalls. This modification may help escaping from the last stationary point obtained and speed up the convergence to an integer feasible solution. A possibility might be that of considering a convex combination of two different merit functions:

$$\phi(x) = \lambda\phi_1(x) + (1-\lambda)\phi_2(x) \tag{37}$$

with $\lambda \in [0, 1]$, and modifying the λ parameter as soon as the algorithm stalls. This is equivalent to use, in the RFP algorithm:

1) a matrix W^k with the following terms:

$$w_j^k = \lambda^k |g_j^k| + (1-\lambda^k)|h_j^k| \quad j = 1, \ldots, n$$

   where $g_j^k \in \partial\phi_1(\bar{x}^k)$ and $h_j^k \in \partial\phi_2(\bar{x}^k)$;

2) an updating rule for the λ parameter that slightly (significantly) changes the penalty term anytime a perturbation (restart) is performed.

In Figure 3 we can see the behaviour of a function obtained by combining the exponential and the logistic function.

7 Numerical Results

In this section we report computational results to compare our version of the FP with the Feasibility Pump algorithm described in [15]. The test set used in our numerical experience consists of 143 instances of 0-1 problems from MIPLIB2003 [2] and COR@L libraries. All the algorithms were implemented in C and we have used ILOG Cplex [24] as solver of the linear programming problems. All tests have been run on an Intel Core2 E8500 system (3.16GHz) with 3.25GB of RAM.
We compare the FP with the reweighted version in two different scenarios:

1 **Randomly generated starting points**: for the terms (8), (22)-(25), we solved the corresponding penalty formulation (21) by means of the Frank-Wolfe algorithm using 1000 randomly generated starting points. The aim of the experiment was to highlight the ability of each penalty formulation to find an integer feasible solution.

2 **FP vs RFP**: in order to evaluate the effectiveness of the new penalty functions, we compared the Feasibility Pump algorithm with the reweighted Feasibility Pump, in which the distance $\Delta_W(x, \tilde{x})$ is defined using the terms (22)-(25).

3 **FP vs Combined RFP**: we made a comparison between the Feasibility Pump algorithm and the reweighted Feasibility Pump with the distance $\Delta_W(x, \tilde{x})$ obtained combining two different penalty terms. The aim of the experiment was to show that the combination of two different functions can somehow improve the RFP algorithm performance.

We performed our experiments using:

- Penalty term (8) denoted by **FP**;
- Penalty term (22) denoted by **Log**, with $\varepsilon = 0.1$;
- Penalty term (23) denoted by **Hyp**, with $\varepsilon = 0.1$;
- Penalty term (24) denoted by **Exp**, with $\alpha = 0.5$;
- Penalty term (25) denoted by **Logis**, with $\alpha = 0.1$.

We stop the algorithms if an integer solution is found or if the limit of 1500 iterations is reached. Due to the random effects introduced by perturbations and major restarts, each problem is tested on a particular penalty function on 10 runs (with different random seeds).

7.1 Computational results for randomly generated starting points

In this first experiment we applied Frank-Wolfe algorithm to solve problem (21) with the objective functions (8), (22)-(25). The algorithm stops when it finds a stationary point (which is not necessarily integer feasible). The goal of the experiment was to understand how good is each function in finding an integer feasible solution. In order to obtain reliable statistics we used 1000 randomly generated starting points. The results obtained on the MIP problems when using randomly generated starting points are shown in Figure 4, where we report the box plots related to the distribution of the number of integer feasible solutions found by each function (we discarded the problems where no function found an integer feasible solution). On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.
We can observe that the results obtained by means of the Exp and the Logis functions, in terms of number of integer feasible solutions found, are slightly better than those obtained using the

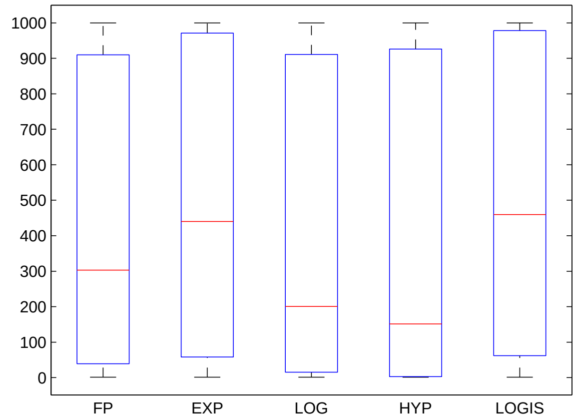


Figure 4: Comparison between the original FP term and the new terms for randomly generated starting points.

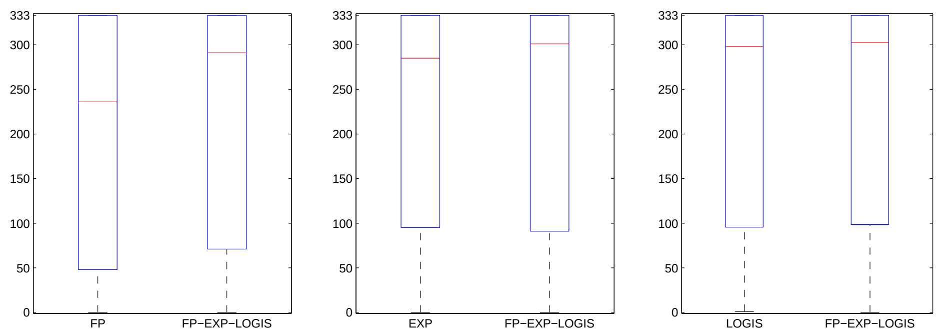


Figure 5: Number of integer feasible solutions found in the parallel experiment.

FP. FP, in turn, guarantees better results than Log and Hyp penalty functions. Anyway, the logarithmic and hyperbolic functions find, for a consistent number of problems, the highest number of integer feasible solutions, so giving a good tool for finding an integer feasible solution. This preliminary computational experience seems to show that the functions have a different behavior in forcing the integrality of the solution. These diversities could be somehow exploited into a multistart strategy. In particular, we could develop a new framework where the minimization of different functions is carried out in parallel. In order to investigate the effect of the parallel use of different functions, we applied the Frank-Wolfe algorithm to three formulations (using three different randomly generated starting points) and we chose the solution with the highest number of integer components among the three. We compared this strategy with the one where we use the same formulation on three different starting points. In Figure 5, we report the results obtained on 333 repetitions of the parallel experiment, when using for each repetition:

- the same formulation with three different starting points;
- three different formulations (FP, Exp and Logis) each one with a different starting point.

We discarded the problems where in both cases no integer feasible solution over the 333 repetitions was found. The results show that using three different formulations is better than using just one formulation. Then, a wider availability of efficient penalty functions is important since it can ease the search of integer feasible solutions for different classes of problems.

7.2 Comparison between FP and RFP

In order to evaluate the ability of finding a first feasible solution, we report in Table 1, for each penalty term:

- The number of problems for which no feasible solution has been found (Not found);
- The number of problems for which a feasible solution has been found at least once (Found at least once);
- The number of problems for which a feasible solution has been found for all the ten runs (Found 10 times);
- The mean number of feasible solutions found (Average number of f.s. found).

As we can easily see from Table 1, FP, Exp and Logis terms have a similar behavior and they are slightly better than Hyp and Log terms.
In Table 2, in order to show the efficiency in terms of objective function value, we report for each penalty term:

- Number of problems for which the best o.f. value (mean over ten runs) is obtained (Best Mean o.f.);
- Number of problems for which the best o.f. value (minimum over ten runs) is obtained (Best Min o.f.).

As we can see by taking a look at Table 2, the Log and Hyp terms guarantee the best performance in terms of objective function value. Furthermore, Exp and Logis terms are comparable and perform better than FP term.

	Not found	Found at least once	Found 10 times	Mean number of f.s. found
FP	16	9	128	8.61
Exp	15	11	127	8.75
Log	18	15	120	8.28
Hyp	27	15	111	7.65
Logis	16	11	126	8.71

Table 1: Comparison between FP and RFP (Feasible solutions)

	Best mean o.f.	Best min o.f.
FP	24	24
Exp	28	27
Log	30	26
Hyp	32	28
Logis	27	25

Table 2: Comparison between FP and RFP (Objective function value)

The detailed results of the comparison between the Feasibility Pump algorithm and the reweighted version obtained using the penalty terms (22)-(25) are shown in Tables 7 - 11. The results related to the problems for which an integer feasible solution is found in all the ten runs are reported in Tables 7 - 9. On the vertical axis of the tables, we have

- the mean number of iterations needed to find a solution (Iter),
- the mean objective function value of the first integer feasible solution found (Obj),
- the mean CPU time (Time).

The results related to the problems for which an integer feasible solution is found in less than ten runs are reported in Tables 10 - 11. On the vertical axis of the tables, we have

- the number of times an integer feasible solution is found (F. s. found),
- the mean number of iterations needed to find a solution (Iter),
- the mean CPU time (Time).

In case of failure, we report "-" for both Iter and Time. By taking a look at the tables, we can notice that the RFP algorithm obtained using the **Exp** penalty (Exp RFP algorithm) and the one obtained using the **Logis** penalty (Logis RFP algorithm) are competitive with the FP in terms of both number of iterations and CPU time. They are also better than the RFP algorithm with the **Log** penalty (Log RFP algorithm) and the one with the **Hyp** penalty (Hyp RFP algorithm) that, in addition, have a larger number of failures. Despite these facts, Log RFP and Hyp RFP algorithms generally give good results in terms of objective function value. In order to better assess the differences in terms of iterations and CPU time between FP and the various versions of the RFP algorithm, we report in Table 3 the geometric means for all the algorithms calculated over 108 instances (those problems for which a feasible solution is found in all the ten runs). In the calculations of the geometric means individual values smaller than 1 are replaced by 1. The results in Table 3 seem to confirm that Exp and Logis RFP algorithms are competitive with FP algorithm.

FP		**Exp**, $\varepsilon = 0.1$		**Log**, $\varepsilon = 0.1$		**Hyp**, $\varepsilon = 0.1$		**Logis**,$\alpha = 0.1$	
Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time
5.774	1.793	4.851	1.683	5.684	1.657	7.193	1.757	4.869	1.678

Table 3: Comparison between FP and RFP (Geometric Means)

In order to better assess the differences between the FP algorithm and the Reweighted FP algorithm, we considered the 126 problems for which an integer feasible solution is found in all the ten runs by FP, Exp RFP and Logis RFP algorithms. We divided the problems in three different classes depending on the CPU time t (seconds) needed by the FP algorithm to solve the problem:

- **Easy.** Problems solved by FP in a time $t \leq 1$ (81 problems);
- **Medium.** Problems solved by FP in a time $1 < t \leq 20$ (34 problems);
- **Hard.** Problems solved by FP in a time $t > 20$ (11 problems).

We report in Figure 6 the results, in terms of CPU time, obtained by the FP, Exp RFP and Logis RFP algorithms on the three classes of problems. Exp RFP and Logis RFP are comparable with FP on the Easy and Medium classes, while they outperform it on the Hard class. Once again, we could develop a new framework where different algorithms are used in parallel. In order to investigate the effect of the parallel use of different algorithms, we ran three algorithms and we chose the solution with the lowest CPU time among the three. We report in Figure 7 the results obtained using:

- 3 runs of the FP algorithm;

- one different algorithm (FP, Exp RFP and Logis RFP) for each run.

By taking a look at the results, we can easily see that the use of different functions improves the performance in Medium and Hard classes, while guarantees comparable results on the Easy class.

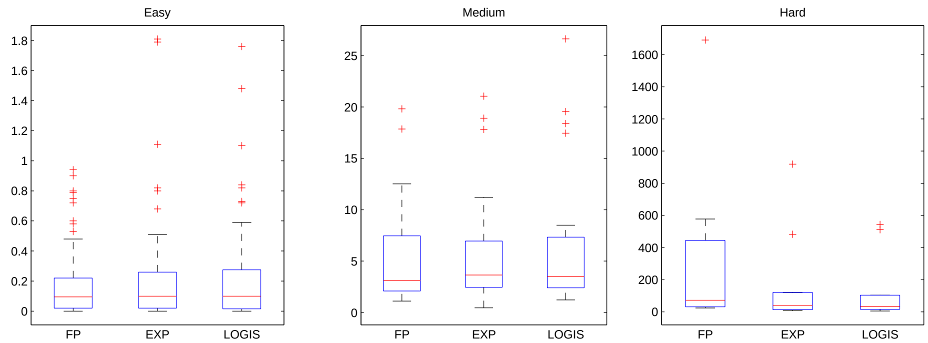


Figure 6: Results in terms of CPU time for the three classes of problems.

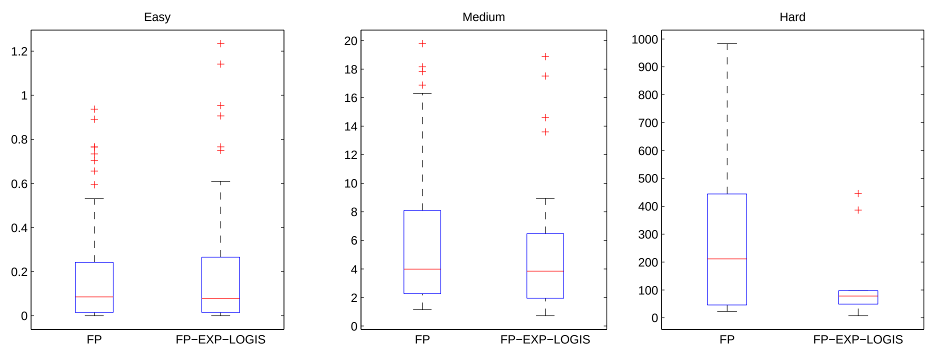


Figure 7: Results in terms of CPU time for the parallel experiment.

7.3 Comparison between FP and combined RFP

In this subsection, we show the effects of combining two different functions. We report the results obtained combining the following functions:

- **Fp** term and **Log**, denoted by **FP+Log**;
- **Exp** term and **Log** term, denoted by **Exp+Log**;
- **Logis** term and **Log** term, denoted by **Logis+Log**;
- **Exp** term and **Logis** term, denoted by **Exp+Logis**.

We set $\psi_1(x)$ equal to the penalty function obtained using the first term and $\psi_2(x)$ equal to the other penalty function. We start with $\lambda^0 = 1$ and we reduce it every time a perturbation occurs. More precisely, we can have two different cases:

- *Weak Perturbation Update:* $\lambda^{k+1} = 0.5\lambda^k$

- *Strong Perturbation Update:* $\lambda^{k+1} = 0.1\lambda^k$

When a strong perturbation occurs, it means that the algorithm is stuck in a cycle. Then the updating rule significantly changes the penalty term, so moving towards the function belonging to the second class.
In order to evaluate the ability of finding a first feasible solution, we report in Table 4, for each penalty term:

- The number of problems for which no feasible solution has been found (Not found);
- The number of problems for which a feasible solution has been found at least once (Found at least once);
- The number of problems for which a feasible solution has been found for all the ten runs (Found 10 times);
- The mean number of feasible solutions found (Average number of f.s. found).

As we can easily see from Table 4, All terms have a similar behavior.
In Table 5, in order to show the efficiency in terms of objective function value, we report for each penalty term:

- Number of problems for which the best o.f. value (mean over ten runs) is obtained (Best Mean o.f.);
- Number of problems for which the best o.f. value (minimum over ten runs) is obtained (Best Min o.f.).

As we can see by taking a look at Table 5, the combined terms guarantee better performance in terms of objective function value than the FP term. Furthermore, Exp+Log combination guarantees the best performance.

	Not solved	Solved at least once	Solved 10 times	Mean number of f.s. found
FP	16	9	128	8.61
FP+Log	17	11	125	8.61
Exp+Log	19	6	128	8.55
Logis+Log	17	9	127	8.58
Exp+Logis	16	10	127	8.59

Table 4: Comparison between FP and Combined RFP (Feasible solutions)

	Best mean o.f.	Best min o.f.
FP	19	19
FP+Log	31	30
Exp+Log	35	33
Logis+Log	32	30
Exp+Logis	32	30

Table 5: Comparison between FP and Combined RFP (Objective function value)

The detailed results of the comparison between the Feasibility Pump algorithm and the reweighted version obtained using the combied penalty terms are shown in Tables 12 - 16. The results related to the problems for which an integer feasible solution is found in all the ten runs are reported in Tables 12 - 14. On the vertical axis of the tables, we have

- the mean number of iterations needed to find a solution (Iter),
- the mean objective function value of the first integer feasible solution found (Obj),
- the mean CPU time (Time).

The results related to the problems for which an integer feasible solution is found in less than ten runs are reported in Tables 15 - 16. On the vertical axis of the tables, we have

- the number of times an integer feasible solution is found (F. s. found),
- the mean number of iterations needed to find a solution (Iter),
- the mean CPU time (Time).

In case of failure, we report "-" for both Iter and Time. By taking a look at the tables, we can notice that the Combined RFP algorithm obtained using the **Exp** and the **Logis** penalty (Exp+Logis RFP algorithm) guarantees the best performance. Furthermore, all the versions of the Combined RFP algorithm are competitive with the standard FP algorithm. We report in Table 6 the geometric means for all the algorithms calculated over 123 instances (those problems for which a feasible solution is found in all the ten runs). In the calculations of the geometric means individual values smaller than 1 are replaced by 1. The results in Table 6 seem to confirm that the Exp+Logis RFP Algorithm is the best among the combined versions of the RFP algorithm and that all the combined RFP algorithms behave favorably when compared to the original FP algorithm in terms of CPU time.

FP		**FP+Log**		**Exp+Log**		**Logis+Log**		**Exp+Logis**	
Iter	Time	Iter	Time	Iter	Time	Iter	Time	Iter	Time
6.252	2.034	6.474	1.630	6.438	1.650	6.388	1.663	5.765	1.617

Table 6: Comparison between FP and Combined RFP (Geometric Means)

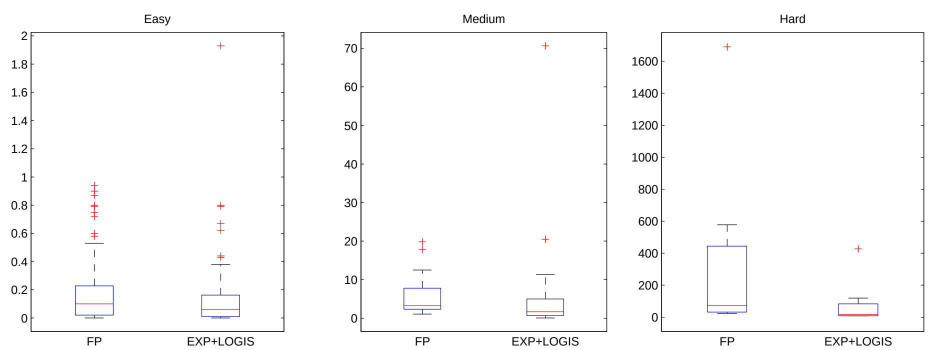


Figure 8: Results in terms of CPU time for the three classes of problems.

In order to better assess the differences between the FP algorithm and the Exp+Logis RFP algorithm, we considered the 124 problems for which an integer feasible solution is found in all the ten runs by the two algorithms. We divided the problems in three different classes depending on the CPU time t (seconds) needed by FP to solve the problem:

- **Easy.** Problems solved by FP in a time $t \leq 1$ (81 problems);
- **Medium.** Problems solved by FP in a time $1 < t \leq 20$ (33 problems);

- **Hard.** Problems solved by FP in a time $t > 20$ (10 problems).

We report in Figure 8 the results, in terms of CPU time, obtained by the FP and Exp+Logis RFP algorithms on the three classes of problems. As we can see, Exp+Logis RFP improves the performance in all the classes.

7.4 Benchmarking Algorithms via Performance Profiles

In order to give a better interpretation of the results generated by the various algorithms we decided to use performance profiles [14]. We consider a set A of n_a algorithms, a set P of n_p problems and a performance measure $m_{p,a}$ (e.g. number of iteration, CPU time). We compare the performance on problem p by algorithm a with the best performance by any algorithm on this problem using the following *performance ratio*

$$r_{p,a} = \frac{m_{p,a}}{\min\{m_{p,a} \ : \ a \in A\}}.$$

Then, we obtain an overall assessment of the performance of the algorithm by defining the following value

$$\rho_a(\tau) = \frac{1}{n_p}\text{size}\{p \in P \ : \ r_{p,a} \leq \tau\},$$

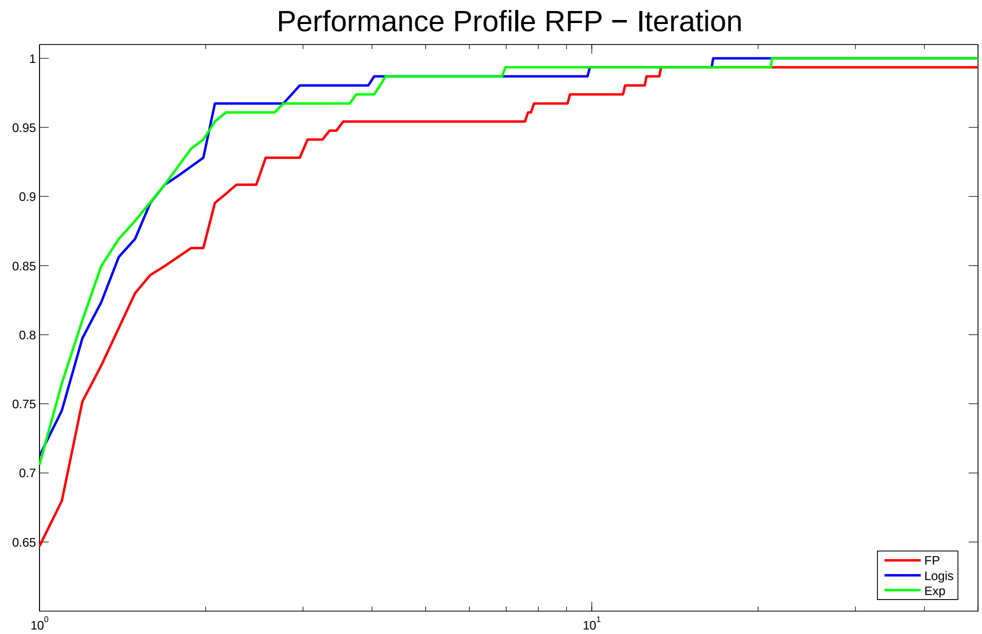
which represents the probability for algorithm $a \in A$ that the performance ratio $r_{p,a}$ is within a factor $\tau \in R$ of the best possible ratio. The function ρ_a represents the distribution function for the performance ratio. Thus $\rho_a(1)$ gives the fraction of problems for which the algorithm a was the most effective, $\rho_a(2)$ gives the fraction of problems for which the algorithm a is within a factor of 2 of the best algorithm, and so on.

In Figure 9, we report the performance profiles related to the comparison between FP, Exp RFP and Logis RFP, in terms of number of iterations and CPU time. It is clear that Exp RFP and Logis RFP functions have a higher number of wins in terms of number of iterations and Exp RFP has the highest number of wins in terms of computational time. Furthermore, the two RFP algorithms are better in terms of robustness.

In Figure 10, we report the performance profiles related to the comparison between FP and the combined version of the RFP obtained using Exp and Logis functions, in terms of number of iterations and CPU time. We can notice that the FP is slightly better in the number of wins, but the combined RFP is better in terms of robustness. The performance profiles related to the CPU time clearly show that the combined RFP outperforms the FP both in terms of number of wins and robustness.

8 Conclusions

In this paper, we focused on the problem of finding a first feasible solution for a 0-1 MIP problem. We started by interpreting the Feasibility Pump heuristic as a Frank-Wolfe method applied to a nonsmooth concave merit function. Then we noticed that the reduction of the merit function used in the FP scheme can correspond to an increase in the number of noninteger variables of the solution. For this reason, we proposed new concave penalty functions that can be included in the FP scheme having two important properties: they decrease whenever the number of integer variables increases; if they decrease, then the number of noninteger variables does not increase. Due to these properties, the functions proposed should speed up the convergence towards integer feasible points. We reported computational results on a set of 143 0-1 MIP problems. This numerical experience shows that the new version of the Feasibility Pump obtained using two of the proposed functions (namely Exp and Logis) compares favorably with the original version



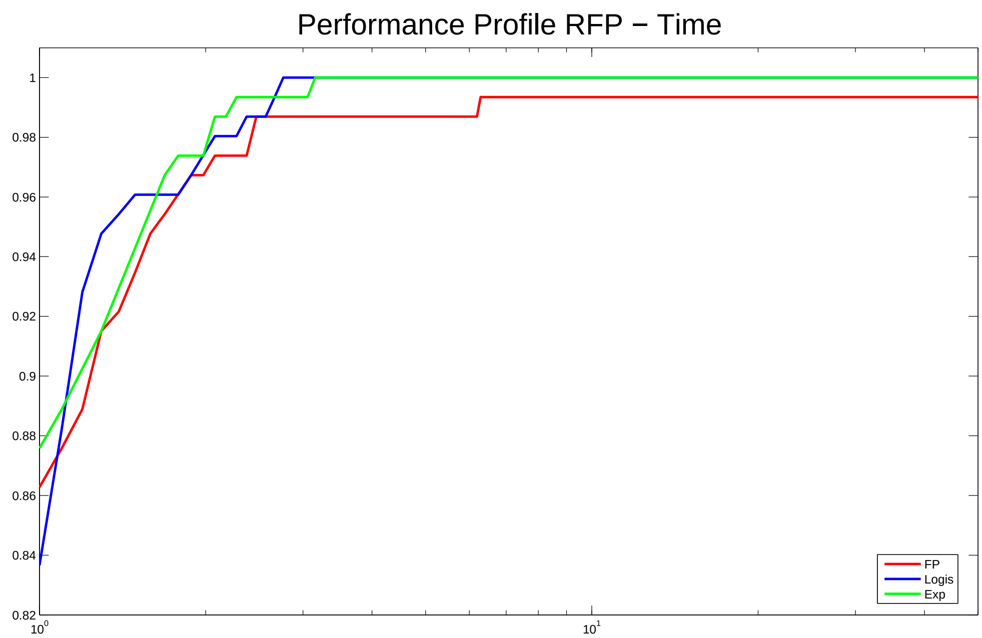
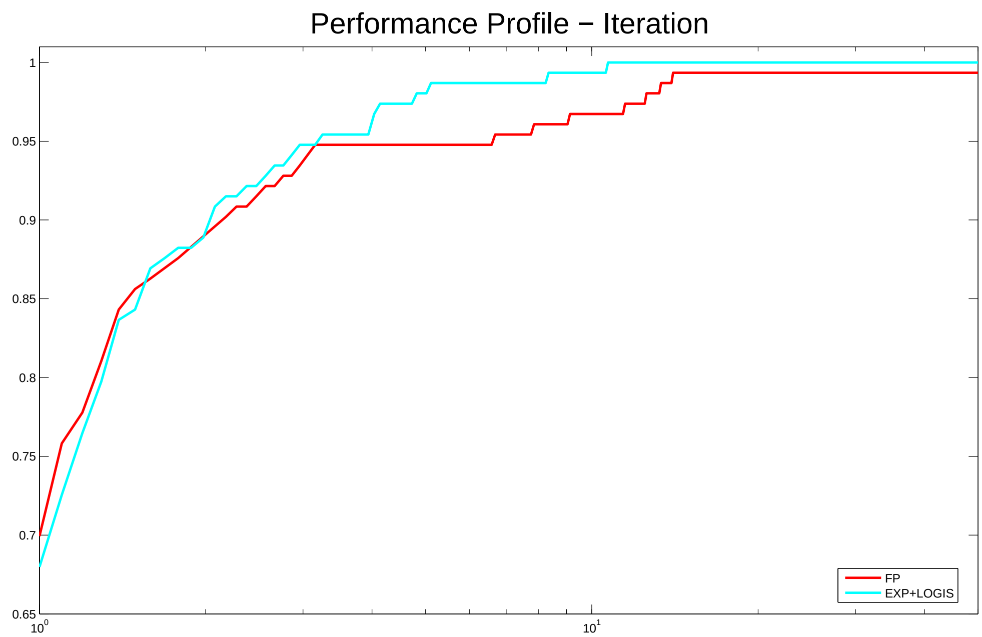


Figure 9: Comparison between FP and the RFP versions (exponential and logistic function). Performance profiles



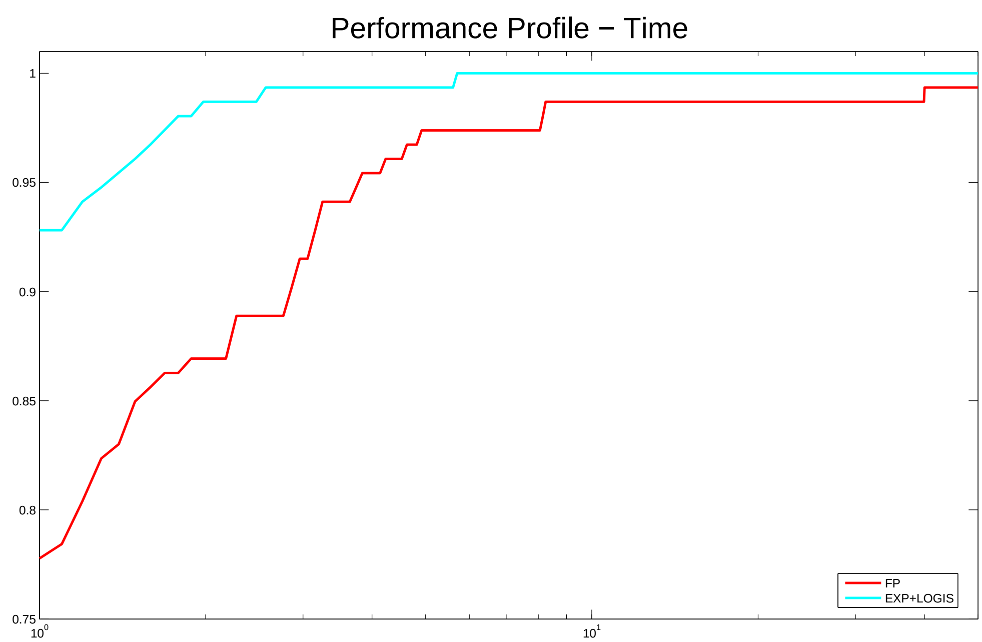


Figure 10: Comparison between FP and the combined RFP version (exponential+logistic function). Performance profiles

of the FP. Furthermore, it highlights that the use of more than one merit function at time (i.e. parallel framework, combination of functions) can significantly improve the efficiency of the algorithm.

In [13], we reinterpret the FP for general MIP problems as a Frank-Wolfe method applied to a suitably chosen function and we extend our approach to this class of problems. In order to not get the present work cumbersome, we decided not to report these results.

Future work will be devoted to the definition of new perturbing procedures based on the proposed functions, to the development of new FP-like methods that, by taking into account the objective function values, guarantee the improvement of the solution quality, and to an extensive numerical experience performed on general MIP problems.

9 Appendix A

For convenience of the reader we report the proof of Proposition 4. We recall a general result concerning the equivalence between an unspecified optimization problem and a parameterized family of problems.

Consider the problems

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in W \end{array} \tag{38}$$

$$\begin{array}{ll} \min & f(x) + \psi(x,\varepsilon) \\ \text{s.t.} & x \in X \end{array} \tag{39}$$

We state the following

Theorem 1 *Let W and X be compact sets. Let $\|\cdot\|$ be a suitably chosen norm. We make the following assumptions.*

A1) *The function f is bounded on X and there exists an open set $A \supset W$ and a real number $L > 0$, such that, $\forall\ x, y \in A$, f satisfies the following condition:*

$$|f(x) - f(y)| \leq L\|x - y\|. \tag{40}$$

The function ψ satisfies the following conditions:

A2) $\forall\ x, y \in W$ *and* $\forall\ \varepsilon \in \mathbb{R}_+$,

$$\psi(x,\varepsilon) = \psi(y,\varepsilon).$$

A3) *There exist a value $\hat{\varepsilon}$ and, $\forall\ z \in W$, there exists a neighborhood $S(z)$ such that, $\forall\ x \in S(z) \cap (X \setminus W)$, and $\varepsilon \in]0, \hat{\varepsilon}]$, we have*

$$\psi(x,\varepsilon) - \psi(z,\varepsilon) \geq \hat{L}\|x - z\|, \tag{41}$$

where $\hat{L} > L$ is chosen as in (40). Furthermore, let $S = \bigcup_{z \in W} S(z)$, $\exists\ \bar{x} \notin S$ such that

$$\lim_{\varepsilon \to 0} [\psi(\bar{x},\varepsilon) - \psi(z,\varepsilon)] = +\infty, \quad \forall\ z \in W, \tag{42}$$

$$\psi(x,\varepsilon) \geq \psi(\bar{x},\varepsilon), \quad \forall\ x \in X \setminus S,\ \forall\ \varepsilon > 0. \tag{43}$$

Then, $\exists\ \tilde{\varepsilon} \in \mathbb{R}$ such that, $\forall\ \varepsilon \in]0, \tilde{\varepsilon}]$, Problems (38) and (39) have the same minimum points.

Proof. See [28].

Now we give the proof of the Proposition 4, with

$$W = \Big\{x \in P : x_i \in \{0,1\},\ \ \forall i \in I\Big\}, \quad X = \Big\{x \in P : 0 \le x_i \le 1,\ \ \forall i \in I\Big\}.$$

Proof of Proposition 4. As we assume that the function f satisfies assumption $A1$) of Theorem 1, the proof can be derived by showing that every penalty term (22)-(25) satisfies assumption $A2$) and $A3$) of Theorem 1.

Consider the penalty term (22).
Let c be the cardinality of I, for any $x \in W$ we have

$$\psi(x, \varepsilon) = c \cdot \log(\varepsilon)$$

and $A2$) is satisfied.
We now study the behavior of the function $\phi(x_i)$, $i \in I$, in a neighborhood of a point $z_i \in \{0,1\}$.
We distinguish three different cases:

1. $z_i = 0$ and $0 < x_i < \rho$ with $\rho < \frac{1}{2}$: We have that $\phi(x_i) = \ln(x_i + \varepsilon)$ which is continuous and differentiable for $0 < x_i < \rho$, so we can use mean value Theorem obtaining that

$$\phi(x_i) - \phi(z_i) = \left(\frac{1}{\tilde{x}_i + \varepsilon}\right)|x_i - z_i|, \tag{44}$$

with $\tilde{x}_i \in (0, x_i)$. Since $\tilde{x}_i < \rho$, we have

$$\phi(x_i) - \phi(z_i) \ge \left(\frac{1}{\rho + \varepsilon}\right)|x_i - z_i|. \tag{45}$$

Choosing ρ and ε such that

$$\rho + \varepsilon \le \frac{1}{\hat{L}}, \tag{46}$$

we obtain

$$\phi(x_i) - \phi(z_i) \ge \hat{L}|x_i - z_i|. \tag{47}$$

2. $z_i = 1$ and $1 - \rho < x_i < 1$ with $\rho < \frac{1}{2}$: We have that $\phi(x_i) = \ln(1 - x_i + \varepsilon)$ which is continuous and differentiable for $1 - \rho < x_i < 1$, so we can use mean value Theorem obtaining that

$$\phi(x_i) - \phi(z_i) = \left(-\frac{1}{1 - \tilde{x}_i + \varepsilon}\right)(x_i - z_i) = \Big(\frac{1}{1 - \tilde{x}_i + \varepsilon}\Big)|x_i - z_i|, \tag{48}$$

with $\tilde{x}_i \in (x_i, 1)$. Since $\rho < \frac{1}{2}$ and $\tilde{x}_i > 1 - \rho$ we have $\frac{1}{1-\tilde{x}_i} > \frac{1}{\rho}$ then

$$\phi(x_i) - \phi(z_i) \ge \left(\frac{1}{\rho + \varepsilon}\right)|x_i - z_i|. \tag{49}$$

We have again that (47) holds when ρ and ε satisfy (46).

3. $z_i = x_i = 0$ or $z_i = x_i = 1$: We have $\phi(x_i) - \phi(z_i) = 0$.

We can conclude that, when ρ and ε satisfy (46),

$$\psi(x,\varepsilon) - \psi(z,\varepsilon) \geq \hat{L} \sum_{i \in I} |x_i - z_i| \geq \hat{L} \sup_{i \in I} |x_i - z_i| \tag{50}$$

for all $z \in W$ and all x such that $\sup_{i \in I} |x_i - z_i| < \rho$.
Now we define $S(z) = \{x \in \mathbf{R}^n : \sup_{i \in I} |x_i - z_i| < \rho\}$ and $S = \bigcup_{i=1}^N S(z_i)$ where N is the number of points $z \in W$.
Let $\bar{x} \notin S$ be such that $\exists j \in I :\ \bar{x}_j = \rho$ $(\bar{x}_j = 1 - \rho)$ and $\bar{x}_i \in \{0, 1\}$ for all $i \neq j$, $i \in I$.
Let $\{\varepsilon^k\}$ be a sequence such that $\varepsilon^k \to 0$ for $k \to \infty$, we can write for each $z \in W$:

$$\begin{aligned} \lim_{k\to\infty} [\psi(\bar{x},\varepsilon^k) - \psi(z,\varepsilon^k)] &= \lim_{k\to\infty} \Big([\ln(\rho + \varepsilon^k) + (c-1)\ln(\varepsilon^k)] - c\ln(\varepsilon^k) \Big) = \\ & \lim_{k\to\infty} \Big(\ln(\rho + \varepsilon^k) - \ln(\varepsilon^k) \Big) = +\infty \end{aligned}$$

and (42) holds.
Then $\forall x \in X \backslash S$, and $\forall \varepsilon > 0$ we have for the monotonicity of the logarithm:

$$\begin{aligned} \psi(x,\varepsilon) - \psi(\bar{x},\varepsilon) &= \sum_{i \neq j} \min\{\ln(x_i + \varepsilon), \ln(1 - x_i + \varepsilon)\} - (c-1)\ln(\varepsilon) \\ &+ \min\{\ln(x_{\bar{j}} + \varepsilon), \ln(1 - x_{\bar{j}} + \varepsilon)\} - \ln(\rho + \varepsilon) \geq 0, \end{aligned}$$

where $\rho \leq x_{\bar{j}} \leq 1 - \rho$. Then (43) holds, and Assumption $A3$) is satisfied.
The proofs of the equivalence between (31) and (32) using the other penalty terms follow by repeating the same arguments used here. □

References

[1] T. Achterberg and T. Berthold. *Improving the feasibility pump.* Discrete Optimization, 4, pp 77–86, 2007.

[2] T. Achterberg, T. Koch, and A. Martin. *MIPLIB 2003.* Operations Research Letters, 34, pp 361–372, 2006. Problems available at http://miplib.zib.de.

[3] E. Balas, S. Ceria, M. Dawande, F. Margot, G. Pataki. *OCTANE: A new heuristic for pure 0-1 programs.* Operations Research, 49(2), pp 207-225, 2001.

[4] E. Balas, C.H. Martin. *Pivot-and-complement: A heuristic for 0-1 programming.* Management Science, 26(1), pp 86-96, 1980.

[5] E. Balas, S. Schmieta, C. Wallace.*Pivot and shifta mixed integer programming heuristic.* Discrete Optimization, 1(1), pp 3-12, 2004.

[6] J. Baxter. , *Local optima avoidance in depot location.* Journal of the Operational Research Society, 32, pp. 815-819, 1981.

[7] L. Bertacco, M. Fischetti, and A. Lodi. *A feasibility pump heuristic for general mixed-integer problems.* Discrete Optimization, 4, pp 63–76, 2007.

[8] J. Eckstein and M. Nediak. *Pivot, cut, and dive: a heuristic for 0-1 mixed integer programming.* Journal of Heuristics, 13, pp 471–503, 2007.

[9] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuejols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N.Sawaya and A. Waechter. *An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs.* IBM Research Report RC23771, To appear in Discrete Optimization, in press.

[10] P. Bonami, G. Cornuejols, and A. Lodi, F. Margot. *A feasibility pump for mixed integer nonlinear programs.* Mathematical Programming, 119, pp 331–352, 2009.

[11] C. D'Ambrosio, A. Frangioni, L. Liberti, A. Lodi. *A Storm of Feasibility Pumps for Nonconvex MINLP.* Mathematical Programming, to appear, 2011.

[12] E. Danna, E. Rothberg, C. Le Pape. *Exploring relation induced neighborhoods to improve MIP solution.* Mathematical Programming 102, 1, pp 71–90, 2005.

[13] M. De Santis, S. Lucidi, F. Rinaldi. *Feasibility Pump-Like Heuristics for Mixed Integer Problems.* DIS Technical Report n. 15, 2010.

[14] E. D. Dolan, J. J. Moré. *Benchmarking optimization software with performance profile.* Mathematical Programming 91, pp 201–213, 2002.

[15] M. Fischetti, F. Glover, A. Lodi. *The Feasibility Pump.* Mathematical Programming, 104, pp 91–104, 2005.

[16] M. Fischetti, A. Lodi. *Local Branching.* Mathematical Programming, 98(1-3), pp 23–47, 2003.

[17] M. Fischetti, D. Salvagnin. *Feasibility pump 2.0.* Mathematical Programming Computation, 1, pp 201–222, 2009.

[18] F. Glover, M. Laguna. *General purpose heuristics for integer programming part I.* Journal of Heuristics, 3, 1997.

[19] F. Glover, M. Laguna. *General purpose heuristics for integer programming part II.* Journal of Heuristics, 3, 1997.

[20] F. Glover, M. Laguna. *Tabu Search.* Kluwer Academic Publisher, Boston, Dordrecht, London, 1997.

[21] F. Glover, A. Løkketangen, D.L. Woodruff. *Scatter search to generate diverse MIP solutions.* in: M. Laguna, J. Gonzàlez-Velarde (Eds.), OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research, Kluwer Academic Publishers, pp. 299317, 2000.

[22] F.S. Hillier, *Efficient heuristic procedures for integer linear programming with an interior.* Operations Research, 17, pp 600-637, 1969.

[23] F.S. Hillier, R.M. Saltzman., *A heuristic ceiling point algorithm for general integer linear programming.* Management Science, 38(2), pp 263-283, 1992.

[24] ILOG, Cplex. http://www.ilog.com/products/cplex.

[25] R. H. Leary. *Global optimization on funneling landscapes.* J. Global Optim., 18, pp. 367–383, 2000.

[26] A. Løkketangen, F. Glover. , *Solving zero/one mixed integer programming problems using tabu search.* European Journal of Operations Research, 106, pp 624-658, 1998.

[27] H.R. Lourenço, O. C. Martin, T. Stülze. , *Iterated local search.* Handbook of metaheuristics - Eds F. W. Glover and G. A. Kochenberger - Kluwer Academic Publishers, Boston, Dordrecht, London, pp. 321-353, 2003.

[28] S. Lucidi, F. Rinaldi. *Exact penalty functions for nonlinear integer programming problems.* Accepted for publication on JOTA.

[29] O. L. Mangasarian. *Solutions of General Linear Complementarity Problems via Nondifferentiable Concave Minimization.* Acta Mathematica Vietnamica, 22(1), pp 199–205, 1997.

[30] O. L. Mangasarian. *Machine learning via polyhedral concave minimization.* in: Fischer, H., Riedmueller, B., Schaeffler S. Applied mathematics and parallel computing- Festschrift for Klaus Ritter, pp 175–188, Physica, Heidelberg, 1996.

[31] W. Murray, K.M. Ng *An Algorithm for Nonlinear Optimization Problems with Binary Variables*, Computational Optimization and Applications, Vol. 47, No. 2, pp 257–288, 2010.

[32] Rinaldi F. *New results on the equivalence between zero-one programming and continuous concave programming*, Optimization Letters, Vol. 3, No. 3, 377–386 (2009).

[33] F. Rinaldi, F. Schoen, M. Sciandrone. *Concave programming for minimizing the zero-norm over polyhedral sets.*, Computational Optimization and Applications, vol. 46, pp. 467,486, 2010.

Problem	**FP**			**Exp** ,	$\varepsilon = 0.1$		**Log**,	$\varepsilon = 0.1$		**Hyp**,	$\alpha = 0.5$		**Logis**,	$\alpha = 0.1$	
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
a1c1s1	25.60	21615.71	2.96	22.70	20365.53	2.37	15.90	20648.03	1.67	26.20	20248.53	2.10	14.70	19736.22	2.56
aflow30a	19.90	5691.70	0.07	12.00	4685.30	0.05	13.40	4049.80	0.04	11.30	3431.10	0.03	14.80	5224.80	0.05
aflow40b	8.50	5711.90	0.12	12.80	5897.20	0.16	11.70	6230.50	0.14	23.30	5934.30	0.21	6.70	4911.10	0.09
cap6000	18.10	-1733965.90	1.20	21.10	-1478513.70	1.50	19.20	-1724741.60	1.01	23.40	-1607966.90	0.94	15.70	-1887299.50	1.22
dano3mip	3.00	1000.00	19.82	1.00	1000.00	18.92	1.00	1000.00	15.51	1.00	1000.00	15.63	2.00	1000.00	26.64
danoint	113.30	87.15	2.78	122.40	87.05	3.69	45.40	85.80	1.09	151.20	88.85	2.84	97.70	87.63	3.35
fast0507	3.00	179.00	97.38	2.00	185.00	98.55	1.00	190.00	85.09	1.00	192.00	86.99	3.00	185.00	103.97
fiber	7.60	14951893.74	0.02	7.90	15096530.88	0.03	9.20	15701099.78	0.03	7.60	12604483.15	0.03	7.60	15472237.20	0.02
fixnet6	11.40	11727.70	0.02	114.00	31150.40	0.25	78.70	24590.60	0.17	64.60	23190.40	0.09	5.40	14731.30	0.01
glass4	25.00	11525809584.00	0.05	105.60	10109632700.00	0.22	73.60	10712142477.00	0.15	258.50	12344594103.00	0.26	100.70	9849049943.00	0.21
harp2	188.80	-47961545.60	1.52	398.00	-48273596.10	3.31	431.40	-41761358.90	3.65	245.00	-46349981.30	2.18	360.20	-44859132.60	3.03
liu	1.00	8398.00	0.09	1.00	8398.00	0.09	1.00	8398.00	0.10	1.00	8398.00	0.09	1.00	8398.00	0.10
markshare1	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00
markshare2	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00
mas74	1.00	19197.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00
mas76	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00
mkc	3.60	-271.65	0.10	3.80	-271.85	0.11	3.70	-271.85	0.11	3.50	-271.85	0.09	3.30	-271.65	0.15
mod011	1.00	0.00	0.07	1.00	0.00	0.07	1.00	0.00	0.10	1.90	3683598.35	0.14	1.00	0.00	0.07
modglob	1.00	602725627.40	0.00	1.00	562873856.90	0.00	1.00	667672982.90	0.00	1.00	663659861.20	0.01	1.00	598677581.40	0.00
net12	42.00	337.00	6.79	153.80	337.00	21.06	142.20	337.00	14.97	113.90	337.00	18.53	117.10	337.00	18.40
nsrand-ipx	3.60	346416.00	0.22	3.20	402048.00	0.27	4.10	355872.00	0.25	5.10	304112.00	0.25	3.10	401408.00	0.27
opt1217	1.00	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	1.00	-12.00	0.01
pk1	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00
pp08aCUTS	3.40	12982.00	0.01	4.00	13104.00	0.01	3.60	11770.00	0.01	3.40	12051.00	0.01	3.90	12581.00	0.01
pp08a	3.10	12810.00	0.00	3.40	13152.00	0.01	3.00	13189.00	0.00	3.00	13615.00	0.00	5.20	13226.00	0.01
qiu	5.60	1539.38	0.19	4.80	1524.65	0.21	5.00	1387.35	0.19	4.40	669.84	0.22	4.30	1687.76	0.28
set1ch	4.20	104900.20	0.01	3.90	101702.80	0.01	33.60	96175.68	0.03	31.10	92687.93	0.03	4.60	105014.45	0.01
seymour	4.00	471.00	2.50	3.00	480.00	2.41	3.00	482.00	1.80	2.00	495.00	1.57	3.00	471.00	2.52
sp97ar	5.20	1468425892.00	5.39	4.50	1722631846.00	6.59	4.00	893925335.90	5.65	4.00	17668956146.00	4.47	4.70	1479783859.00	6.86
swath	84.80	36527.08	7.11	61.30	28614.67	5.37	61.90	35450.07	5.23	566.80	48160.31	36.22	34.30	21903.21	3.51
tr12-30	83.70	243560.80	0.22	154.20	260762.20	0.45	62.50	260330.90	0.20	114.10	247401.30	0.31	114.30	245892.50	0.30
vpm2	5.80	23.88	0.00	4.00	20.93	0.01	5.00	20.65	0.00	3.50	19.25	0.00	5.30	20.25	0.00

Table 7: Comparison on MIPLIB problems (integer feasible solution found in all the ten runs). FP vs RFP

Problem	FP			Exp ,	$\varepsilon = 0.1$		Log,	$\varepsilon = 0.1$		Hyp,	$\alpha = 0.5$		Logis,	$\alpha = 0.1$	
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
22433	8.50	21527.40	0.06	12.20	21527.50	0.07	13.20	21509.00	0.09	50.80	21550.30	0.14	7.80	21550.80	0.07
23588	51.60	8310.40	0.11	46.80	8314.90	0.12	392.30	8316.40	0.80	380.90	8293.70	0.77	69.40	8325.00	0.17
bc1	2.20	12.90	0.58	2.40	15.34	0.51	2.90	13.36	0.46	2.70	15.59	0.44	2.40	16.24	0.59
bienst1	11.40	89.92	0.09	1.00	68.25	0.06	1.00	68.25	0.07	1.00	68.25	0.07	1.30	75.93	0.10
bienst2	13.30	127.10	0.12	1.00	68.25	0.07	1.00	68.25	0.06	1.00	68.25	0.08	1.00	102.22	0.10
binkar10-1	27.20	609256.29	0.15	26.70	909412.54	0.15	31.40	608918.49	0.17	74.70	1509667.48	0.40	29.40	1009533.69	0.17
dano3-3	12.50	1000.00	31.74	1.00	1000.00	13.29	1.00	996.08	16.48	1.00	758.11	11.58	1.20	997.24	17.73
dano3-4	7.80	1000.00	23.95	1.00	1000.00	13.61	1.00	1000.00	13.25	1.00	1000.00	13.17	1.00	974.74	15.88
dano3-5	9.10	997.67	26.46	1.00	1000.00	14.59	1.00	1000.00	14.88	1.00	1000.00	14.83	1.00	1000.00	16.52
mcf2	146.70	82.97	3.67	85.20	85.70	2.62	100.30	86.50	2.38	183.30	86.85	3.65	173.70	82.70	6.06
mkc1	1.00	-460.93	0.12	1.00	-146.86	0.08	1.00	-311.19	0.15	1.00	-289.23	0.07	1.00	-525.33	0.12
neos5	1.00	21.00	0.00	1.00	21.00	0.00	1.00	22.00	0.01	1.00	21.00	0.00	1.00	22.00	0.01
neos6	11.80	141.60	3.50	20.00	146.80	4.80	191.30	157.40	20.42	540.50	158.40	49.85	34.60	142.20	6.97
neos13	1.00	-28.43	1.29	1.00	0.00	0.72	1.00	0.00	0.64	1.00	-37.43	0.75	1.00	-13.14	1.27
neos14	5.50	215724354.30	0.03	6.40	237071334.90	0.03	6.40	253618668.00	0.03	5.00	275922098.90	0.02	5.00	247318443.50	0.03
neos17	2.60	0.68	0.04	2.60	0.66	0.04	2.60	0.61	0.04	2.60	0.61	0.04	2.60	0.75	0.04
neos18	1.00	36.00	0.13	2.00	34.00	0.14	20.60	37.80	0.70	39.00	40.50	1.12	2.00	34.00	0.13
neos-430149	137.70	497.95	0.79	177.10	499.60	0.82	423.10	498.66	1.76	356.30	539.58	1.42	118.40	516.19	0.72
neos-476283	3.00	1056.42	444.74	1.00	729.57	121.23	1.00	681.38	116.94	1.00	630.09	152.10	1.00	680.77	71.75
neos-480878	3.00	590.70	0.10	3.00	624.72	0.10	3.00	546.81	0.08	3.00	556.93	0.09	3.00	610.31	0.11
neos-494568	2.00	29.00	1.48	1.00	-74.00	2.96	2.00	-83.00	1.45	4.90	238.70	1.88	1.00	26.00	1.67
neos-504674	85.80	30961.35	0.25	45.90	29748.56	0.14	56.00	29114.83	0.17	83.80	30126.00	0.25	11.30	29898.73	0.05
neos-504815	82.40	13912.75	0.20	118.30	15388.38	0.29	82.10	13813.72	0.20	158.00	15177.66	0.38	164.80	14854.18	0.40
neos-512201	191.20	5373.11	0.53	171.80	5165.76	0.50	210.00	5248.57	0.62	160.20	5270.02	0.48	198.80	5287.04	0.58
neos-522351	6.40	103262.07	0.48	4.90	38323.98	0.34	4.70	32313.14	0.30	6.40	31111.00	0.26	5.90	86648.70	0.58
neos-525149	1.00	61.00	12.01	1.00	63.00	11.22	1.00	63.00	9.88	1.00	66.00	7.89	1.00	63.00	7.34
neos-538867	60.40	6425.00	0.33	70.60	6072.50	0.43	53.80	8814.00	0.23	124.90	9108.50	0.50	58.30	6242.00	0.34
neos-538916	38.20	5650.00	0.20	24.90	5955.80	0.16	102.40	7938.10	0.46	160.60	7922.10	0.70	35.80	6139.80	0.20
neos-547911	18.40	15.30	7.81	5.20	15.60	3.25	12.80	15.80	3.41	15.60	15.60	2.39	9.70	14.70	6.10
neos-555694	9.00	55.90	0.35	4.00	24.80	0.21	8.50	90.39	0.33	28.10	106.77	0.57	66.30	108.52	1.10
neos-555771	56.00	130.84	1.10	17.10	91.99	0.45	45.80	123.10	0.87	16.10	95.79	0.38	169.00	110.21	2.64
neos-565815	1.00	14.00	9.12	2.00	14.00	10.13	55.00	14.70	24.38	62.20	14.80	23.44	1.00	14.00	7.32
neos-570431	4.70	27.00	0.27	5.40	37.70	0.32	4.20	14.30	0.23	5.00	19.30	0.17	5.00	29.70	0.29
neos-584851	4.00	-4.00	0.04	2.80	-3.90	0.04	39.90	-3.30	0.14	87.10	-2.50	0.31	3.80	-4.80	0.05
neos-603073	8.00	47327.85	0.08	10.40	46853.05	0.13	5.00	46611.70	0.06	5.00	46550.72	0.06	9.30	46704.76	0.11
neos-611838	4.00	4849174.32	2.18	3.00	4342869.45	1.91	3.40	4102837.86	2.93	3.50	4309473.45	2.45	3.40	5043085.61	2.16
neos-612125	3.00	4792546.67	2.81	4.80	4232007.52	3.64	4.60	4378948.77	4.26	4.10	4364312.31	3.17	3.00	4793407.04	1.83
neos-612143	3.00	4805355.24	2.92	5.20	4139911.87	2.44	4.10	4166564.73	1.80	4.20	4408307.64	3.14	3.00	4598667.63	1.90

Table 8: Comparison on COR@L problems (integer feasible solution found in all the ten runs). FP vs RFP - Part I

Problem	**FP**			**Log**,	$\varepsilon = 0.1$		**Hyp**,	$\varepsilon = 0.1$		**Exp**,	$\alpha = 0.5$		**Logis**,	$\alpha = 0.1$	
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
neos-612162	3.40	4827358.83	2.93	3.40	4436190.19	3.16	4.90	4252915.18	2.71	4.50	4311181.25	3.46	3.00	5166641.34	1.99
neos-655508	0.00	63015042.00	0.04	0.00	63015042.00	0.04	0.00	63015042.00	0.04	0.00	63015042.00	0.04	0.00	63015042.00	0.04
neos-775946	124.10	764.30	3.25	126.40	857.61	3.26	41.90	714.14	1.87	80.80	749.25	1.98	98.80	794.52	2.47
neos-780889	2.00	10821585.00	48.19	2.40	11040195.00	52.38	2.00	10967417.50	50.68	2.80	11258515.00	63.65	2.00	10906187.50	50.17
neos-801834	2.00	64502.00	0.80	1.00	54872.00	0.40	2.00	61289.00	0.36	2.00	60964.00	0.38	2.00	62990.00	0.84
neos-824695	3.70	77.00	0.75	3.70	77.00	0.80	3.90	77.00	0.85	3.70	77.00	0.82	4.10	77.00	0.82
neos-825075	4.00	218.00	0.06	8.00	544.00	0.10	3.00	8.00	0.06	198.30	903.00	0.92	3.00	218.00	0.06
neos-826250	3.10	63.00	0.40	3.30	63.00	0.44	3.30	63.00	0.42	3.30	63.00	0.42	3.10	63.00	0.38
neos-826812	2.70	83.01	0.72	2.80	83.01	0.68	2.70	83.01	0.66	2.80	83.01	0.69	2.70	83.01	0.73
neos-827175	2.00	121.00	1.80	2.00	121.00	2.24	2.00	121.00	2.23	2.00	121.00	2.24	2.00	121.00	1.81
neos-839859	1.00	94247985.64	0.20	1.00	131658548.10	0.21	1.00	58556618.20	0.20	1.00	58556618.20	0.21	1.00	131658548.10	0.21
neos-860300	14.30	7685.30	3.13	13.70	8203.30	2.45	25.60	7092.90	2.03	144.90	9005.70	4.52	10.70	6677.80	2.85
neos-886822	2.00	138398.00	0.26	1.00	178597.50	0.20	1.00	28820.50	0.17	1.00	28820.50	0.16	2.00	178597.50	0.25
neos-892255	3.60	18.70	0.15	3.70	18.90	0.14	3.80	18.80	0.13	10.80	48.40	0.33	3.70	18.90	0.14
neos-906865	2.00	9105.20	0.05	2.00	9910.60	0.05	2.00	9910.20	0.05	2.00	10714.80	0.04	2.00	10712.40	0.05
neos-955215	2.20	9037.66	0.01	3.00	967.60	0.01	3.00	911.58	0.01	3.00	897.42	0.01	3.40	928.70	0.01
neos-1058477	2.80	3.58	0.02	2.40	3.76	0.02	2.80	2.78	0.02	2.40	3.74	0.03	3.80	5.40	0.03
neos-1171448	1.00	0.00	0.60	1.00	0.00	0.50	1.00	0.00	0.53	1.00	0.00	0.45	1.00	0.00	0.49
neos-1200887	1.00	-38.00	0.02	1.00	-52.00	0.02	1.00	-42.00	0.02	1.00	-38.00	0.02	1.00	-44.00	0.02
neos-1211578	1.00	-51.00	0.00	1.00	-48.00	0.01	1.00	-44.00	0.00	1.00	-52.00	0.00	1.00	-48.00	0.00
neos-1225589	27.20	23555348134.00	0.05	10.60	24272916822.00	0.02	29.30	23041323223.00	0.06	26.20	23771592587.00	0.05	16.60	22423855518.00	0.03
neos-1228986	1.00	-92.00	0.00	1.00	-80.00	0.00	1.00	-72.00	0.01	1.00	-70.00	0.01	1.00	-75.00	0.00
neos-1337489	1.00	-51.00	0.00	1.00	-48.00	0.01	1.00	-44.00	0.00	1.00	-52.00	0.00	1.00	-48.00	0.00
neos-1413153	2.00	119.12	0.37	1.00	119.12	0.39	1.00	119.12	0.38	1.00	119.12	0.40	1.00	119.12	0.37
neos-1415183	1.00	425.60	0.53	1.00	128.61	0.46	1.00	128.61	0.48	1.00	128.61	0.47	1.00	425.60	0.58
neos-1437164	23.60	25.90	0.14	63.80	23.30	0.35	22.50	22.70	0.13	42.00	21.00	0.25	9.20	23.50	0.06
neos-1440447	1.00	-52.00	0.01	1.00	-56.00	0.01	1.00	-60.00	0.01	1.00	-46.00	0.01	1.00	-60.00	0.01
neos-1460265	35.70	15925.00	0.18	17.40	15820.00	0.11	25.90	15910.00	0.15	40.80	15840.00	0.21	28.80	15905.00	0.16
neos-1480121	2.00	89.33	0.00	2.00	95.80	0.00	2.00	95.80	0.00	2.00	95.80	0.00	2.00	89.33	0.00
neos-1489999	5.80	476.90	0.05	6.80	483.00	0.06	5.10	498.30	0.05	4.70	487.40	0.05	6.20	481.50	0.05
neos-1516309	9.00	54363.50	0.13	12.70	53987.00	0.17	11.70	53707.00	0.15	9.80	54282.00	0.13	11.80	53105.00	0.15
neos-1595230	3.50	20.40	0.10	4.10	21.30	0.10	4.50	20.50	0.11	4.90	21.80	0.10	4.00	20.30	0.10
neos-1597104	4.60	-7.10	8.08	4.30	-7.50	11.10	8.70	-2.00	8.98	28.50	-7.50	30.90	4.00	-6.50	8.50
neos-1599274	3.00	36277.60	0.17	3.00	37547.60	0.17	6.00	37347.60	0.14	17.20	52419.12	0.38	5.30	53258.16	0.17
neos-1620807	8.80	9.50	0.02	7.20	9.50	0.02	10.50	9.80	0.02	7.80	9.80	0.02	7.00	9.10	0.02
prod1	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
qap10	516.80	502.40	1690.54	1.00	406.00	7.45	1.00	406.00	7.91	2.00	406.00	11.95	1.00	406.00	8.74
roy	38.30	5810.25	0.03	30.30	5887.40	0.02	17.20	5761.75	0.02	30.70	5806.61	0.03	37.20	5590.45	0.03

Table 9: Comparison on COR@L problems (integer feasible solution found in all the ten runs). FP vs RFP - Part II

Problem	**FP**			**Exp** ,	$\varepsilon = 0.1$		**Log**,	$\varepsilon = 0.1$		**Hyp**,	$\alpha = 0.5$		**Logis**,	$\alpha = 0.1$	
	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time
10teams	10	122.40	7.34	10	107.00	6.24	1	-	-	0	-	-	10	92.30	5.86
air04	10	11.20	12.52	10	4.60	8.13	5	-	-	0	-	-	10	21.20	19.56
air05	10	2.00	2.42	10	3.00	2.88	10	7.00	4.58	1	-	-	10	5.00	3.51
misc07	10	39.60	0.13	10	62.90	0.20	10	490.80	0.96	8	-	-	10	46.50	0.16
momentum1	10	474.20	577.99	10	382.40	482.49	5	-	-	0	-	-	10	450.70	544.05
nw04	10	1.00	0.94	10	1.00	1.79	8	-	-	7	-	-	10	1.00	1.48
p2756	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
protfold	10	360.20	107.67	9	-	-	0	-	-	0	-	-	10	553.50	162.36
t1717	10	18.00	366.80	10	56.40	918.95	5	-	-	1	-	-	10	24.10	511.38

Table 10: Comparison on MIPLIB problems (feasible solution found in less than ten runs). FP vs RFP

Problem	FP			Exp , $\varepsilon = 0.1$			Log, $\varepsilon = 0.1$			Hyp, $\alpha = 0.5$			Logis, $\alpha = 0.1$		
	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time
aligninq	10	380.10	6.01	10	621.70	9.54	2	-	-	0	-	-	8	-	-
lrn	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos2	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos3	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos11	10	5.30	0.90	10	14.40	1.81	10	111.80	4.56	8	-	-	10	14.70	1.76
neos12	10	5.00	7.80	10	5.00	8.02	6	724.00	154.67	0	-	-	10	6.00	8.28
neos-583731	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-593853	1	-	-	10	69.70	0.93	7	-	-	0	-	-	6	-	-
neos-598183	10	91.70	0.87	9	-	-	10	71.40	0.65	6	-	-	10	83.30	0.78
neos-631694	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-709469	4	-	-	3	-	-	0	-	-	0	-	-	3	-	-
neos-777800	10	13.70	5.19	10	16.90	6.52	10	54.70	12.67	2	-	-	10	4.00	1.98
neos-791021	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-799711	0	-	-	10	83.80	659.80	10	1.30	194.87	10	26.70	198.20	9	-	-
neos-799716	0	-	-	9	-	-	9	-	-	7	-	-	4	-	-
neos-803219	0	-	-	0	-	-	2	-	-	5	-	-	0	-	-
neos-803220	5	-	-	9	-	-	10	253.00	1.55	10	183.30	1.15	9	-	-
neos-806323	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-807639	2	-	-	2	-	-	1	-	-	1	-	-	0	-	-
neos-807705	0	-	-	0	-	-	0	-	-	2	-	-	0	-	-
neos-810286	10	139.10	46.72	10	90.30	29.90	3	-	-	0	-	-	10	10.00	5.77
neos-810326	10	668.10	76.05	6	-	-	0	-	-	0	-	-	9	-	-
neos-820879	10	5.00	1.68	10	19.10	4.79	10	47.30	8.69	6	-	-	10	11.00	3.86
neos-829552	10	1.00	17.86	10	2.00	17.82	10	33.50	52.66	1	-	-	10	1.00	17.46
neos-862348	9	-	-	9	-	-	10	415.30	4.34	10	259.40	2.95	8	-	-
neos-880324	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-912015	6	-	-	5	-	-	0	-	-	0	-	-	6	-	-
neos-932816	2	-	-	2	-	-	1	-	-	0	-	-	0	-	-
neos-941698	10	29.80	0.80	10	48.80	1.11	10	435.00	5.78	0	-	-	10	47.40	1.10
neos-948268	10	5.00	6.36	10	6.00	6.16	10	9.00	9.07	2	-	-	10	7.00	7.93
neos-957270	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-957389	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-1215259	7	-	-	5	-	-	0	-	-	0	-	-	8	-	-
neos-1281048	10	131.80	1.79	10	338.60	4.60	7	-	-	0	-	-	10	173.00	2.65
neos-1396125	2	-	-	0	-	-	5	-	-	4	-	-	2	-	-
neos-1441553	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-

Table 11: Comparison on COR@L problems (feasible solution found in less than ten runs). FP vs RFP

Problem	**FP**			**FP -**	**Log**		**Exp -**	**Log**		**Logis -**	**Log**		**Exp -**	**Logis**	
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
10teams	122.40	994.40	7.34	231.90	975.80	13.09	197.60	1008.60	9.48	192.40	998.00	10.84	201.00	1007.20	11.37
a1c1s1	25.60	21615.71	2.96	30.10	22509.27	2.20	38.00	22635.76	2.74	24.90	23436.21	2.12	20.80	22357.30	2.05
aflow30a	19.90	5691.70	0.07	10.10	3636.30	0.01	7.50	3309.30	0.01	8.40	3747.80	0.02	8.20	4176.30	0.01
aflow40b	8.50	5711.90	0.12	6.70	4085.40	0.04	9.20	4663.10	0.05	8.00	4581.20	0.05	7.20	4962.50	0.04
air04	11.20	61461.90	12.52	28.60	69078.20	65.59	19.30	70144.90	45.23	9.80	59614.30	23.25	28.20	68491.40	70.62
air05	2.00	32368.00	2.42	13.80	36682.50	16.54	21.00	36708.80	24.20	40.70	44456.90	27.54	3.00	29948.00	6.04
cap6000	18.10	-1733965.90	1.20	28.80	-1799142.90	1.01	17.60	-1755643.20	0.67	8.20	-1735379.30	0.35	6.20	-2007735.40	0.29
dano3mip	3.00	1000.00	19.82	1.00	1000.00	10.83	1.00	1000.00	10.17	1.00	1000.00	10.44	1.00	1000.00	10.64
danoint	113.30	87.15	2.78	173.80	82.00	3.40	80.10	83.28	1.76	180.20	84.00	3.60	111.00	82.10	2.46
fast0507	3.00	179.00	97.38	1.00	186.00	7.37	1.00	186.00	6.63	1.00	186.00	9.94	3.00	195.00	23.47
fiber	7.60	14951893.74	0.02	6.20	9512215.27	0.02	6.20	9512708.95	0.02	8.00	14195745.63	0.02	7.00	14123680.53	0.02
fixnet6	11.40	11727.70	0.02	113.70	27806.50	0.12	143.70	31703.80	0.15	114.50	29716.30	0.12	121.90	27972.00	0.13
glass4	25.00	11525809584.00	0.05	76.40	8157128425.00	0.05	107.00	7479410064.00	0.07	89.40	7453308715.00	0.06	102.70	7863699741.00	0.07
liu	1.00	8398.00	0.09	1.00	4720.00	0.06	1.00	4720.00	0.06	1.00	4720.00	0.06	1.00	4720.00	0.07
markshare1	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00	1.00	292.00	0.00
markshare2	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00	1.00	160.00	0.00
mas74	1.00	1917.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00	1.00	19197.47	0.00
mas76	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00	1.00	44877.42	0.00
misc07	39.60	4236.50	0.13	75.10	4388.50	0.10	75.20	4442.50	0.11	58.90	4251.50	0.09	67.20	4237.00	0.10
mkc	3.60	-271.65	0.10	3.50	-271.85	0.10	3.40	-271.85	0.08	3.50	-271.85	0.10	3.40	-271.85	0.10
mod011	1.00	0.00	0.07	1.00	0.00	0.04	1.00	0.00	0.04	1.00	0.00	0.04	1.00	0.00	0.03
modglob	1.00	602725627.40	0.00	1.00	625805480.20	0.00	1.00	562041867.10	0.00	1.00	560046949.70	0.00	1.00	625805480.20	0.00
net12	42.00	337.00	6.79	133.20	337.00	4.20	109.30	337.00	3.32	197.10	337.00	5.56	168.80	337.00	4.88
nsrand-ipx	3.60	346416.00	0.22	4.00	345600.00	0.28	3.20	378336.00	0.30	4.20	347168.00	0.31	3.20	393680.00	0.29
nw04	1.00	19882.00	0.94	5.00	19657.00	25.29	12.80	29484.60	64.31	11.20	42121.80	50.59	1.00	19882.00	0.43
opt1217	1.00	0.00	0.01	1.00	-12.00	0.01	1.00	-12.00	0.01	1.00	-12.00	0.00	1.00	-12.00	0.01
pk1	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00	1.00	36.00	0.00
pp08aCUTS	3.40	12982.00	0.01	3.70	12223.00	0.01	3.70	12030.00	0.01	3.70	12208.00	0.01	4.60	12624.00	0.01
pp08a	3.10	12810.00	0.00	3.00	12505.00	0.00	3.00	12453.00	0.00	3.00	12439.00	0.00	3.80	12949.00	0.00
qiu	5.60	1539.38	0.19	3.80	1390.67	0.13	4.30	1491.29	0.13	5.10	954.68	0.13	4.80	1741.11	0.13
set1ch	4.20	104900.20	0.01	6.00	83983.00	0.01	6.00	83983.00	0.01	8.90	83247.70	0.01	4.00	84122.05	0.01
seymour	4.00	471.00	2.50	3.00	482.00	1.05	3.00	482.00	1.07	2.00	481.00	1.06	3.00	477.00	1.12
sp97ar	5.20	1468425892.00	5.39	6.00	876827395.70	1.39	4.00	998491944.60	1.36	4.00	934439637.40	1.40	7.00	1606141803.00	1.66
swath	84.80	36527.08	7.11	69.60	36824.53	3.12	69.20	33118.44	3.32	64.10	27368.93	3.13	61.90	29633.73	3.17
t1717	18.00	201829.70	366.80	69.70	512750.40	373.76	65.50	525868.20	386.00	59.20	363772.90	256.13	52.00	417769.40	427.11
tr12-30	83.70	243560.80	0.22	55.80	262726.40	0.09	83.60	265720.40	0.14	113.20	263093.10	0.17	98.90	271277.70	0.15
vpm2	5.80	23.88	0.00	6.20	20.38	0.00	6.60	19.75	0.00	6.20	20.23	0.00	6.40	22.10	0.00

Table 12: Comparison on MIPLIB problems (integer feasible solution found in all the ten runs). FP vs Combined RFP

Problem	**FP**			**FP -**	**Log**		**Exp -**	**Log**		**Logis -**	**Log**		**Exp -**	**Logis**	
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
22433	8.50	21527.40	0.06	16.40	21540.30	0.05	12.70	21548.30	0.04	12.20	21517.00	0.04	11.40	21541.80	0.04
23588	51.60	8310.40	0.11	95.50	8322.60	0.14	39.80	8301.80	0.06	61.90	8287.70	0.09	35.10	8325.00	0.05
bc1	2.20	12.90	0.58	2.10	9.44	0.17	2.40	10.28	0.17	2.00	10.66	0.17	2.10	9.93	0.17
bienst1	11.40	89.92	0.09	1.00	83.92	0.05	1.00	68.25	0.06	1.00	68.25	0.06	1.00	68.25	0.06
bienst2	13.30	127.10	0.12	1.00	76.03	0.05	1.00	78.00	0.05	1.00	74.00	0.06	1.00	72.42	0.05
binkar10-1	27.20	609256.29	0.15	20.90	408583.99	0.05	20.70	508570.60	0.05	35.90	508968.41	0.08	25.60	608858.49	0.06
dano3-3	12.50	1000.00	31.74	1.00	641.91	8.47	1.00	623.32	8.46	1.00	627.64	8.48	1.00	653.48	8.67
dano3-4	7.80	1000.00	23.95	1.00	651.90	8.63	1.00	674.14	8.51	1.00	668.04	8.53	1.00	665.13	8.65
dano3-5	9.10	997.67	26.46	1.00	691.88	8.62	1.00	709.29	8.72	1.00	706.27	8.74	1.00	670.24	8.62
mcf2	146.70	82.97	3.67	118.80	85.70	2.39	136.30	85.80	3.03	103.80	83.85	2.12	115.20	84.45	2.58
mkc1	1.00	-460.93	0.12	1.00	-566.15	0.03	1.00	-566.15	0.03	1.00	-566.15	0.04	1.00	-566.15	0.03
neos5	1.00	21.00	0.00	2.00	18.00	0.00	1.00	17.00	0.00	2.00	17.50	0.00	2.00	18.00	0.00
neos6	11.80	141.60	3.50	5.30	129.00	0.87	23.40	131.80	2.33	17.50	149.40	2.06	31.40	142.90	2.92
neos11	5.30	10.00	0.90	6.90	9.10	0.84	7.80	9.00	0.74	8.70	9.00	0.77	14.80	10.60	1.93
neos12	5.00	20.00	7.80	39.00	19.50	35.79	20.20	16.20	10.34	41.70	20.60	20.00	4.00	19.00	7.25
neos13	1.00	-28.43	1.29	1.00	-14.95	0.86	1.00	-15.04	0.86	1.00	-16.47	0.78	1.00	-46.34	0.48
neos14	5.50	215724354.30	0.03	4.70	243613848.30	0.01	5.00	281907988.40	0.01	5.00	291385913.70	0.01	5.20	265456557.80	0.01
neos17	2.60	0.68	0.04	2.60	0.58	0.03	2.60	0.58	0.03	2.30	0.54	0.04	2.80	0.58	0.03
neos18	1.00	36.00	0.13	7.20	32.90	0.17	8.30	29.10	0.17	10.00	33.30	0.22	2.00	34.00	0.07
neos-430149	137.70	497.95	0.79	132.30	438.86	0.27	219.00	465.04	0.39	162.20	522.29	0.29	174.40	533.29	0.30
neos-476283	3.00	1056.42	444.74	1.00	523.17	8.48	1.00	511.24	8.37	1.00	514.88	8.48	1.00	541.27	11.13
neos-480878	3.00	590.70	0.10	3.60	542.04	0.04	3.50	540.16	0.03	3.60	553.33	0.04	3.30	562.90	0.03
neos-494568	2.00	29.00	1.48	2.00	-82.00	0.22	2.00	-82.00	0.22	2.00	-81.00	0.22	1.00	-72.00	0.23
neos-504674	85.80	30961.35	0.25	124.60	30946.73	0.15	35.00	31121.71	0.06	80.80	31777.31	0.12	31.10	29473.78	0.05
neos-504815	82.40	13912.75	0.20	96.10	13720.27	0.11	32.90	13982.71	0.05	103.40	15708.03	0.13	54.30	13976.53	0.07
neos-512201	191.20	5373.11	0.53	157.60	5557.65	0.19	165.30	5458.20	0.29	134.10	5407.88	0.23	193.20	5524.32	0.24
neos-522351	6.40	103262.07	0.48	5.30	40010.80	0.07	5.80	46605.30	0.08	4.70	30080.06	0.07	4.70	49141.50	0.08
neos-525149	1.00	61.00	12.01	1.00	65.00	1.60	1.00	65.00	1.61	1.00	65.00	1.47	1.00	63.00	1.46
neos-538867	60.40	6425.00	0.33	82.20	6830.00	0.23	70.30	5419.50	0.19	50.20	5645.00	0.12	69.70	6989.50	0.19
neos-538916	38.20	5650.00	0.20	30.70	6109.20	0.08	49.30	6398.70	0.12	31.70	5846.60	0.08	36.30	6430.40	0.09
neos-547911	18.40	15.30	7.81	12.60	15.00	1.15	15.70	15.00	2.20	11.50	15.40	2.60	7.30	15.30	1.70
neos-555694	9.00	55.90	0.35	16.20	78.56	0.18	17.70	87.49	0.21	17.00	61.18	0.21	4.00	25.00	0.09
neos-555771	56.00	130.84	1.10	17.60	86.74	0.20	11.60	104.41	0.15	16.40	90.83	0.19	4.00	43.60	0.09
neos-565815	1.00	14.00	9.12	8.30	14.50	2.36	5.20	14.80	2.22	9.30	15.40	3.07	5.10	14.20	2.41
neos-570431	4.70	27.00	0.27	4.30	16.00	0.12	3.70	15.10	0.11	3.70	14.80	0.12	5.50	23.80	0.16
neos-584851	4.00	-4.00	0.04	9.50	-5.50	0.04	10.60	-5.20	0.04	12.30	-6.30	0.04	2.50	-4.10	0.03
neos-598183	91.70	48288.78	0.87	18.20	47013.60	0.06	218.30	47841.88	0.46	16.40	47547.14	0.06	135.10	49824.98	0.29
neos-603073	8.00	47327.85	0.08	5.70	46725.08	0.02	5.80	46760.97	0.02	5.50	46171.88	0.02	38.30	49371.71	0.09
neos-611838	4.00	4849174.32	2.18	6.20	3730351.88	0.75	5.20	3645759.07	0.66	5.80	3811374.06	0.82	3.00	3577497.64	0.69
neos-612125	3.00	4792546.67	2.81	5.70	4097819.37	0.97	4.30	3998758.87	0.82	4.20	3928947.03	0.70	3.70	4068154.44	0.93
neos-612143	3.00	4805355.24	2.92	5.90	3838432.32	0.76	5.90	3848910.24	0.69	3.80	3911342.26	0.63	4.00	3666814.85	0.74
neos-612162	3.40	4827358.83	2.93	5.80	3681085.94	0.73	5.90	3927845.17	0.73	4.20	3834411.42	0.61	3.10	3533947.53	0.52
neos-655508	0.00	63015042.00	0.04	0.00	63015042.00	0.03	0.00	63015042.00	0.02	0.00	63015042.00	0.02	0.00	63015042.00	0.03

Table 13: Comparison on COR@L problems (integer feasible solution found in all the ten runs). FP vs Combined RFP - Part I

Problem	FP			FP - Log			Exp - Log			Logis - Log			Exp - Logis		
	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time	Iter	Obj	Time
neos-775946	124.10	764.30	3.25	14.60	391.98	0.50	22.00	530.29	0.61	4.50	531.10	0.39	18.80	496.09	0.57
neos-777800	13.70	-80.00	5.19	4.00	-80.00	1.37	11.30	-80.00	5.14	19.90	-80.00	9.23	10.20	-80.00	6.81
neos-780889	2.00	10821585.00	48.19	2.10	10032625.00	100.65	3.30	10260442.50	96.59	2.00	10178090.00	93.20	2.10	10034935.00	83.28
neos-801834	2.00	64502.00	0.80	2.00	55577.00	0.37	2.00	60875.00	0.37	2.00	61233.00	0.37	1.00	54051.00	0.38
neos-810286	139.10	3431.90	46.72	81.30	3435.40	44.06	74.20	3377.30	44.79	83.30	3316.10	42.20	114.10	3485.80	80.34
neos-820879	5.00	34433.70	1.68	10.70	38492.40	1.32	12.40	37749.10	1.40	13.50	37945.10	1.61	6.50	37208.20	0.98
neos-824695	3.70	77.00	0.75	3.80	77.00	0.64	3.70	77.00	0.63	3.90	77.00	0.65	3.90	77.00	0.67
neos-825075	4.00	218.00	0.06	9.00	465.00	0.06	4.00	108.00	0.04	3.00	8.00	0.04	6.20	395.00	0.04
neos-826250	3.10	63.00	0.40	3.20	63.00	0.35	3.30	63.00	0.38	3.40	63.00	0.37	3.40	63.00	0.38
neos-826812	2.70	83.01	0.72	2.70	83.01	0.59	2.40	83.01	0.56	2.80	83.01	0.62	2.70	83.01	0.62
neos-827175	2.00	121.00	1.80	2.00	121.00	1.12	2.00	121.00	1.12	2.00	121.00	1.13	2.00	121.00	1.14
neos-829552	1.00	26.69	17.86	7.00	2.91	24.91	5.20	2.92	24.53	11.60	42.40	32.62	2.00	6.67	20.52
neos-839859	1.00	94247985.64	0.20	1.00	58556618.20	0.18	1.00	58556618.20	0.17	1.00	58556618.20	0.17	1.00	131658548.10	0.18
neos-860300	14.30	7685.30	3.13	15.90	7321.60	0.87	21.60	7044.40	1.16	17.10	6285.70	0.93	9.70	8861.70	0.73
neos-886822	2.00	138398.00	0.26	1.00	28820.50	0.17	1.00	28820.50	0.16	1.00	28820.50	0.16	1.00	178597.50	0.27
neos-892255	3.60	18.70	0.15	3.90	18.90	0.09	8.00	45.60	0.18	3.90	20.60	0.10	11.50	46.30	0.25
neos-906865	2.00	9105.20	0.05	2.00	10823.90	0.03	2.00	10819.70	0.03	2.00	11060.30	0.03	2.00	9744.10	0.03
neos-941698	29.80	22.30	0.80	48.40	10.00	0.55	98.20	10.40	1.02	64.50	8.30	0.73	62.30	10.20	0.79
neos-948268	5.00	60.00	6.36	13.70	60.00	12.52	6.00	60.00	6.28	7.00	60.00	6.52	3.00	60.00	5.44
neos-955215	2.20	9037.66	0.01	3.00	809.42	0.01	3.00	809.35	0.01	3.00	808.92	0.01	3.40	1029.15	0.01
neos-1058477	2.80	3.58	0.02	2.00	1.47	0.01	2.80	1.46	0.01	3.20	11.00	0.02	4.40	31.25	0.02
neos-1171448	1.00	0.00	0.60	1.00	0.00	0.26	1.00	0.00	0.26	1.00	0.00	0.26	1.00	0.00	0.28
neos-1200887	1.00	-38.00	0.02	1.00	-52.00	0.02	1.00	-52.00	0.01	1.00	-52.00	0.01	1.00	-52.00	0.02
neos-1211578	1.00	-51.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00
neos-1225589	27.20	23555348134.00	0.05	43.60	25241744868.00	0.08	30.00	23484230439.00	0.06	51.30	27161430594.00	0.10	31.40	23827160202.00	0.06
neos-1228986	1.00	-92.00	0.00	1.00	-104.00	0.00	1.00	-104.00	0.00	1.00	-104.00	0.00	1.00	-104.00	0.00
neos-1281048	131.80	173712.90	1.79	243.00	174774.20	1.90	285.60	183703.90	2.08	167.70	175805.50	1.31	308.40	180675.60	2.25
neos-1337489	1.00	-51.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00	1.00	-69.00	0.00
neos-1413153	2.00	119.12	0.37	1.00	119.12	0.35	1.00	119.12	0.35	1.00	119.12	0.35	1.00	119.12	0.36
neos-1415183	1.00	425.60	0.53	1.00	128.61	0.43	1.00	128.61	0.43	1.00	128.61	0.43	1.00	128.61	0.44
neos-1437164	23.60	25.90	0.14	37.30	17.60	0.19	21.10	18.90	0.11	19.70	19.00	0.10	28.70	19.40	0.15
neos-1440447	1.00	-52.00	0.01	1.00	-77.00	0.01	1.00	-79.00	0.00	1.00	-78.00	0.00	1.00	-78.00	0.00
neos-1460265	35.70	15925.00	0.18	175.10	15410.00	1.03	91.70	15490.00	0.51	108.20	15520.00	0.61	143.20	15520.00	0.80
neos-1480121	2.00	89.33	0.00	2.00	95.80	0.00	2.00	95.80	0.00	2.00	95.80	0.00	2.00	96.60	0.00
neos-1489999	5.80	476.90	0.05	6.90	484.30	0.05	6.30	481.60	0.05	6.20	488.30	0.05	6.20	483.50	0.05
neos-1516309	9.00	54363.50	0.13	11.90	54069.00	0.12	12.40	52941.00	0.13	10.80	52827.00	0.12	17.70	53687.50	0.16
neos-1595230	3.50	20.40	0.10	3.80	20.50	0.07	5.00	21.00	0.07	4.70	22.10	0.07	3.70	20.50	0.07
neos-1597104	4.60	-7.10	8.08	8.20	-2.60	1.07	8.20	-2.60	1.05	6.00	-3.40	1.11	4.60	-6.90	0.98
neos-1599274	3.00	36277.60	0.17	8.60	52367.76	0.13	9.20	51694.48	0.14	9.40	51652.80	0.14	3.00	37687.60	0.07
neos-1620807	8.80	9.50	0.02	10.60	9.70	0.02	7.00	9.10	0.02	9.20	9.70	0.02	6.90	9.70	0.01
prod1	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
qap10	516.80	502.40	1690.54	1.00	406.00	7.33	1.00	406.00	10.21	1.00	406.00	7.37	1.00	406.00	10.64
roy	38.30	5810.25	0.03	212.00	5788.85	0.08	98.10	5393.15	0.04	264.00	5622.95	0.10	319.80	5878.40	0.12

Table 14: Comparison on COR@L problems (integer feasible solution found in all the ten runs). FP vs Combined RFP - Part II

Problem	FP			FP - Log			Exp - Log			Logis - Log			Exp - Logis		
	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time	F.s. found	Iter	Time
harp2	10	188.80	1.52	10	525.90	4.28	9	-	-	9	-	-	10	257.40	2.19
momentum1	10	474.20	577.99	9	-	-	10	215.80	56.10	9	-	-	10	578.20	118.96
p2756	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
protfold	10	360.20	107.67	9	-	-	10	524.90	89.36	10	361.60	83.31	9	-	-

Table 15: Comparison on MIPLIB problems (integer feasible solution found in less than ten runs). FP vs Combined RFP

Problem	**FP**			**FP - Log**			**Exp - Log**			**Logis - Log**			**Exp - Logis**		
	F.s found	Iter	Time	F.s found	Iter	Time	F.s found	Iter	Time	F.s found	Iter	Time	F.s found	Iter	Time
aligninq	10	380.10	6.01	10	623.90	3.68	9	-	-	8	-	-	7	-	-
lrn	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos2	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos3	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-583731	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-593853	1	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-631694	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-709469	4	-	-	3	-	-	2	-	-	4	-	-	7	-	-
neos-791021	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-799711	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-799716	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-803219	0	-	-	0	-	-	0	-	-	1	-	-	1	-	-
neos-803220	5	-	-	8	-	-	10	258.00	0.53	10	273.70	0.60	10	275.60	0.55
neos-806323	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-807639	2	-	-	2	-	-	2	-	-	2	-	-	1	-	-
neos-807705	0	-	-	0	-	-	0	-	-	0	-	-	2	-	-
neos-810326	10	668.10	76.05	8	-	-	10	773.30	109.01	10	366.50	59.75	9	-	-
neos-862348	9	-	-	9	-	-	10	65.50	0.86	10	180.40	2.15	10	62.80	0.81
neos-880324	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-912015	6	-	-	7	-	-	5	-	-	2	-	-	2	-	-
neos-932816	2	-	-	4	-	-	0	-	-	2	-	-	2	-	-
neos-957270	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-957389	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
neos-1215259	7	-	-	8	-	-	1	-	-	5	-	-	5	-	-
neos-1396125	2	-	-	1	-	-	0	-	-	0	-	-	0	-	-
neos-1441553	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-

Table 16: Comparison on COR@L problems (feasible solution found in less than ten runs). FP vs Combined RFP