Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 4, No. 1, pp. 75 – 86, January 2008

ELIMINATING DUALITY GAP IN INTEGER PROGRAMMING VIA OBJECTIVE CUTS*

Wei Wang and Yifan Xu†

Dedicated to Professor Liansheng Zhang on the occasion of his 70th birthday.

Abstract: In this paper, we propose an approach which combines the augmented Lagrangian method with objective cuts to successfully guarantee the dual search in generating an optimal solution of a finite integer optimization problem with multiple constraints. Compared to the general nonlinear Lagrangian methods, the proposed methods do not destroy the structure of the original problem. Some numerical results are presented to show the algorithm process.

Key words: *integer programming, duality gap, Lagrangian relaxation, cut*

Mathematics Subject Classification: *90C10, 49M29*

1 Introduction

Consider the following finite integer programming problem:

$$(P) \qquad \min\{f(x) \mid g(x) \leq b,\ x \in X\}, \tag{1.1}$$

where $X \subset R^n$ is a finite integer set, $f : R^n \to R$, $g = (g_1, ..., g_m)^T : R^n \to R^m$ and $b = (b_1, ..., b_m)^T \geq 0$. Without loss of generality, f and g_i, $i = 1, ..., m$, are assumed to be strictly positive for all $x \in X$. Define F to be the feasible region of the problem (P),

$$F = \{x \in X \mid g(x) \leq b\}. \tag{1.2}$$

Denote by $v(Q)$ the optimal value of an optimization problem (Q). Thus, the optimal value of (P) is $v(P)$.

The concept of duality plays a significant role in integer optimization. Incorporating the set of constraints into the objective function yields a Lagrangian relaxation, e.g., Geoffrion (1974), Fisher (1981), and Shapiro (1979). Duality theory and methods have been further investigated in linear integer programming by various authors, e.g., Guignard and Kim (1993), Llewellyn and Ryan (1993), and Williams (1996). Lagrangian dual methods have also

*This research was supported by the National Natural Sciences Foundations of China under No. 70471012, and No.70771028, and the National Basic Research Program of China (973 Program) under No.2007CB814904.

†The corresponding author

been used in some special classes of nonlinear integer programming problems, for instance, Michelon and Maculan (1991,1993). Mathematically, an augmented Lagrangian relaxation problem is defined by

$$(R_{\lambda,r}) \qquad \min_{x\in X} L(x,\lambda,r)=f(x)+\lambda^T(g(x)-b)+\sum_{i=1}^{m} r_i \max\{0,g_i(x)-b_i\}, \qquad (1.3)$$

where $\lambda \in R^m_+$ is the multiplier and $r \in R^m_+$ is the penalty parameter. The Lagrangian dual is an optimization problem in λ,

$$(D) \qquad \max_{\lambda\in R^m_+} v(R_{\lambda,r}).$$

The Lagrangian method searches for an optimal solution of (P) via maximizing the dual function $v(R_{\lambda,r})$. If $\hat{x}$ solves both (P) and $(R_{\hat{\lambda},\hat{r}})$ with $\hat{\lambda},\hat{r} \in R^m_+$ and $\hat{\lambda}$ solves the dual problem (D), then $\hat{\lambda}$ is said to be an optimal generating Lagrangian multiplier vector, $(\hat{x},\hat{\lambda})$ to be an optimal primal-dual pair of (P).

While the Lagrangian method is a powerful constructive dual search method, it often fails to identify an optimal solution of the primal integer optimization problem. There are two situations that would prevent the Lagrangian method from success in the dual search. Firstly, the optimal solution of (P) may not even be generated by solving $(R_{\lambda,r})$ for any $\lambda \geq 0$. Secondly, the optimal solution to $(R_{\lambda^*,r})$, with λ^* being a solution to the dual problem (D), is not necessarily an optimal solution to (P), or even infeasible. The first situation is associated with the existence of an optimal generating Lagrangian multiplier vector. The second situation is related to the existence of an optimal primal-dual pair. To guarantee the successful dual search, Li and Sun (2000) developed the pth power Lagrangian method, which gives a revised version of the t-norm surrogate formulation and then makes a pth power transformation both to the objective function and the single-constraint. Furthermore, Sun and Li (2000) proposed a logarithmic-exponential dual formulation for problem (P), which possesses an asymptotically strong duality and guarantees the identification of an optimal solution of problem (P) by using a nonlinear Lagrangian function. Although these approaches are effective to guarantee the success of dual searches, they may destroy some good structures of the primal integer optimization problem. Li, Wang and Sun (2007) introduced the idea of using objective cuts in reducing duality gap in integer programming where dynamic programming method is employed to solve the Lagrangian relaxation subproblems

In this paper, we develop a method to guarantee the success of dual search without destroying the structure of the objective function and constraints in problem (P). The idea is to construct an auxiliary integer programming which is the combination of original problem (P) and the objective function cuts. By solving the augmented Lagrangian relaxation of the auxiliary problem and corresponding dual problem, we obtain the optimal solution of (P). Since the augmented Lagrangian function can be linearized by Cohen and Zhu (1984), this method can be used to solve large scale problem and some problems with special structures. In each iteration of the method, we solve the augmented Lagrangian relaxation problem of this auxiliary problem. Then the parameters, upper cuts, lower cuts, Lagrangian multipliers and penalty parameters are updated according to whether or not the solution of the relaxation problem is feasible, g-infeasible or b-infeasible. We prove that the optimal solution of problem (P) can be obtained after finite iterations. Numerical results also illustrate that the proposed algorithms are successful in guaranteeing the dual search.

This paper is organized as follows. In section 2, we introduce the auxiliary problem associated with the original problem (P), its augmented Lagrangian relaxation and their perturbation functions. In section 3, we present three algorithms based on cut and discuss the convergence of these algorithms. Finally in section 4, we show the method is effective via some numerical testes .

2 Auxiliary Problem

Let us introduce an auxiliary problem $(P_{(l,u)})$ with respect to problem (P):

$$(P_{(l,u)}) \qquad \min\{f(x) \mid g(x) \leq b,\ x \in X_{(l,u)}\}, \tag{2.1}$$

where

$$X_{(l,u)} = \{x \in X \mid l < f(x) < u\}.$$

The problem $(P_{(l,u)})$ is called an *auxiliary problem with cuts*. l is called the *lower cut* and u the *upper cut*. The *augmented Lagrangian relaxation problem* associated with $(P_{(l,u)})$ is defined as:

$$(R_{\lambda,r,(l,u)}) \qquad \min_{x \in X_{(l,u)}} L(x,\lambda,r) = f(x) + \lambda^T(g(x)-b) + \sum_{i=1}^{m} r_i \max\{0, g_i(x) - b_i\}. \tag{2.2}$$

The *perturbation function* and the *augmented perturbation function* of problem $(P_{(l,u)})$, denoted by $\phi_{(l,u)}(\cdot)$ and $\phi'_{(l,u)}(\cdot)$ respectively, are defined as:

$$\phi_{(l,u)}(y) = \begin{cases} \inf\{f(x) \mid g(x) \leq y,\ x \in X_{(l,u)}\}, & y \in F_{(l,u)}, \\ +\infty, & otherwise \end{cases} \tag{2.3}$$

and

$$\phi'_{(l,u)}(y) = \begin{cases} \inf\{f_r(x) \mid g(x) \leq y,\ x \in X_{(l,u)}\}, & y \in F_{(l,u)}, \\ +\infty, & otherwise, \end{cases} \tag{2.4}$$

where $f_r(x) = f(x) + \sum_{i=1}^m r_i \max\{0, g_i(x) - b_i\}$ and $F_{(l,u)} = \{y \in R_+^m \mid \phi_{(l,u)}(y) < +\infty\}$. From geometry, the graph of perturbation function $\phi_{(l,u)}(\cdot)$ is

$$G_{\phi_{(l,u)}} = \{(y, y_0) \mid y \in F_{(l,u)},\ y_0 = \phi_{(l,u)}(y)\} \tag{2.5}$$

and the epi-graph of $\phi_{(l,u)}(\cdot)$ is

$$Epi_{\phi_{(l,u)}} = \{(y, y_0) \mid y \in F_{(l,u)},\ y_0 \geq \phi_{(l,u)}(y)\}. \tag{2.6}$$

For simplicity, we denote $\phi_{(0,+\infty)}, X_{(0,+\infty)}$ and $F_{(0,+\infty)}$ by ϕ, X and F respectively. A point $(y, y_0) \in \mathrm{Epi}_\phi$ is called a *noninferior point* if and only if $(y', y_0) \notin Epi_\phi$ for any $y' < y$. The set of all noninferior points of Epi_ϕ is denoted by E^0.

It is easy to see that following properties for perturbation function $\phi_{(l,u)}(\cdot)$ and the augmented perturbation function $\phi'_{(l,u)}(\cdot)$ are true:

$\phi_{(l,u)}(\cdot),\ \phi'_{(l,u)}(\cdot)$ are nonincreasing piecewise-constant functions.

$\phi_{(l,u)}(\cdot),\ \phi'_{(l,u)}(\cdot)$ are continuous from right.

$\phi_{(l,u)}(\cdot) \leq \phi'_{(l,u)}(\cdot)$ and $\phi_{(l,u)}(\cdot) = \phi'_{(l,u)}(\cdot)$ for $y \leq b$.

Lemma 2.1. *For any $y \in F$, there exists $\hat{x} \in X$ such that $g(\hat{x}) \le y, \phi(y) = \phi(g(\hat{x}))$ and $(g(\hat{x}), f(\hat{x})) \in E^0$.*

Proof. From the finiteness of X, we assume $X' = \{x_i \mid x_i \in X, g(x_i) \le y\} = \{x_1, ..., x_N\}$ and let $g = (g_1, ..., g_m)^T$, where $g_i = \max\{g_i(x_1), ..., g_i(x_N)\}$ for $i = 1, ..., m$. We search $\hat{x} \in X'$ in following way.

Firstly, we look for a point $x' \in X$ which satisfies $g(x') \le y$ and $\phi(y) = \phi(g(x'))$. If there exists $x_j \in X'$ such that $g(x_j) = g$, then set $x' = x_j$, which meets $\phi(y) = \phi(g(x'))$. Otherwise, suppose that there are $i \ne j$ such that $g_i = g_i(x_N), g_j = g_j(x_{N-1})$ and $x_N \ne x_{N-1}$. If $f(x_N) \ge f(x_{N-1})$, we delete x_N from X', i.e., we set $X' = \{x_1, ..., x_{N-1}\}$. Repeating this process, we can find x' in $k \le N$ steps.

Furthermore, if there exists another point $x'' \in \{x \in X : g(x) \le y\}$ such that $f(x'') < f(x')$, then, from the way that x' is obtained, we have $0 < g(x'') < g(x')$ and $x'' \in \{x_1, ..., x_{N-k}\}$. By the same method searching for x', we finally can obtain such $\hat{x}$ which satisfies $\phi(y) = \phi(g(\hat{x})) = f(\hat{x})$ and $(g(\hat{x}), f(\hat{x})) \in E^0$. We finish the proof. □

Lemma 2.2 (Li and Sun (2000)).
(i) *For any $y \in F$, if $\hat{x}$ is the solution of (2.3), then $(g(\hat{x}), f(\hat{x})) \in G_\phi$.*
(ii) *For any $(a', c') \in E^0$, there exists $x^* \in X$ such that $(a', c') = (g(x^*), f(x^*))$.*

In view of Lemma 2.1 and Lemma 2.2, we know that there exists $x^* \in \{x \in X \mid g(x) \le b\}$ which solves problem (P) and $(g(x^*), f(x^*)) \in E^0$. Since there is no convexity assumption in problem (P), when we solve the augmented Lagrangian relaxation method, the tangent plane can not guarantee to touch the optimal solution of problem (P). Hence, we need the method to expose the point $(f(x^*), g(x^*))$.

Lemma 2.3. *Suppose that $\hat{x}$ is the solution of problem $(R_{\lambda,r})$ with $\lambda \ge 0$ and $r > 0$.*
(i) *If $g(\hat{x}) \le b$, then $(g(\hat{x}), f(\hat{x})) \in E^0$.*
(ii) *If $g(\hat{x}) > b$, then $f(\hat{x}) < f(x^*)$, where x^* is the optimal solution of (P).*

Proof. For the first part, suppose that $(g(\hat{x}), f(\hat{x})) \in Epi_\phi$, but $(g(\hat{x}), f(\hat{x})) \notin E^0$. Then, by (2.3), (2.4) and (2.5), there exists an $x' \in X$ such that $f(x') \le f(\hat{x})$ and $g(x') < g(\hat{x})$. Hence, by $g(\hat{x}) \le b$ and $\lambda > 0$, we have $L(x', \lambda, r) < L(\hat{x}, \lambda, r)$. It is a contradiction to $\hat{x}$ being the solution of (2.2).

For the second part, suppose x^* be the solution of (P). Then, since $\hat{x}$ solve problem $(R_{\lambda,r})$ for $\lambda > 0,\ r > 0$ and , we have

$$f(\hat{x}) + \lambda^T(g(\hat{x}) - b) + \sum_{i=1}^{m} r_i \max\{0, g_i(\hat{x}) - b_i)\} \le f(x^*) + \lambda^T(g(x^*) - b).$$

By $g(\hat{x}) > b$ and $g(x^*) - b \le 0$, we have

$$f(\hat{x}) < f(x^*),$$

which ends the proof. □

Lemma 2.4. *There is a $\hat{r} > 0$ such that for any $r \ge \hat{r}$, if $\hat{x}$ solves $(R_{\lambda,r})$ for given λ, $\hat{x}$ is a feasible solution of (P).*

Proof. In view of the finiteness of X, let

$$f^- = \max\{f(x) \mid x \in X, g(x) \le b\},$$

$$f_- = \min\{f(x)|\ x \in X, g(x) \not\leq b\},$$
$$\alpha = \min_{i=1,\dots,m} \{g_i(x) - b_i > 0|\ x \in X, g(x) \not\leq b\}$$

and

$$\beta = \min_{x \in X} \lambda^T(g(x) - b).$$

Then, we only need to take $\hat{r} = (\hat{r}_1, ..., \hat{r}_m)$, where $\hat{r}_i, i = 1, ..., m$, satisfies

$$f_- + \beta + \hat{r}_i\alpha > f^-,$$

or

$$\hat{r}_i > \frac{f^- - f_- - \beta}{\alpha}.$$

We finish the proof. □

When we solve the augmented Lagrangian relaxation problem, the solution obtained may be feasible or infeasible for problem (P), in particular, *g-infeasible* ($x \in X, g(x) > b$) or *b-infeasible* ($x \in X, g(x) \not\leq b, g(x) \not> b$). From Lemma 2.3 and Lemma 2.4, we can observe that if $\hat{x}$, the solution of $(R_{\lambda,r})$, is feasible, then the optimal solution of (P) is in $\{x \in X|\ f(x) \leq f(\hat{x})\}$; if $\hat{x}$ is g-infeasible, then the optimal solution of (P) is in $\{x \in X|\ f(x) > f(\hat{x})\}$; last if $\hat{x}$ is b-infeasible, then we can increase parameter r to push the solution of $(R_{\lambda,r})$ back into the feasible domain of (P).

Lemma 2.5. *Suppose x^* is the optimal solution of (P). Then,*

(i) *there exists $u > f(x^*)$ such that, for any $z \in (f(x^*), u)$ and suitable r , $Epi_{\phi'_{(0,z)}}$ has a supporting plane at $(g(x^*), f(x^*))$ or*

(ii) *there exists $l < f(x^*)$ such that, for any $z \in (l, f(x^*))$ and suitable r, $Epi_{\phi'_{(z,+\infty)}}$ has a supporting plane at $(g(x^*), f(x^*))$.*

Proof. By the finiteness of X, $\{x \in X \mid f(x) > f(x^*)\}$ is finite. Thus, there exists $u > f(x^*)$ such that $\{x \in X \mid f(x^*) < f(x) < u\} = \emptyset$. We can observe that for given $k > 1$, there exists sufficiently large r such that the plane Π with normal vector $(1, \lambda^T)^T$, where $\lambda_i = f(x^*)/(kb_i - g_i(x^*)), i = 1, ..., m$, supports $Epi_{\phi_{(0,z)}}$ at $(f(x^*), g(x^*))$. Now we show such r is existing. From the finiteness of X again, we have $\min\{g_j(x) - b_j | g_j(x) > b_j, j = 1, 2, ..., m, x \in X\} > 0$. So, for sufficiently large r, point $(g(x), f_r(x))$ where $x \in \{X | g(x) \not\leq b\}$ locates over the plane Π. we finish the proof for (i). The proof for (ii) is similar with those for (i). □

Lemma 2.6. *Suppose x^* is the optimal solution of (P) and l, u and r are defined as those in Lemma 2.5. Then,*

(i) *for any $z \in (f(x^*), u)$, there exists λ such that x^* is an optimal solution of $(R_{\lambda,r,(0,z)})$.*

(ii) *For any $z \in (l, f(x^*))$, there exists λ such that x^* is an optimal solution of $(R_{\lambda,r,(z,+\infty)})$.*

Proof. From Lemma 2.5, there exists plane

$$\Pi = \{(\xi, y) \mid \xi = f(x^*) - \lambda^T(y - g(x^*))\},$$

which is a supporting hyperplane of $\mathrm{Epi}_{\phi'_{(0,u)}}$ at $(f(x^*), g(x^*))$. Therefore, from (2.4), we have

$$\phi'_{(0,u)}(y) \geq \xi(y) = f(x^*) - \lambda^T(y - g(x^*)), \quad \forall y,$$

or in other words,

$$\phi'_{(0,u)}(y) + \lambda(y-b) \geq f(x^*) + \lambda(g(x^*) - b), \quad \forall y.$$

For arbitrary $x \in X_{(0,u)}$, setting $y = g(x)$, we have $g(x) \in F_{(0,u)}$ and $\phi'_{(0,u)}(g(x)) \leq f_r(x)$. So,

$$f(x) + \lambda^T(g(x)-b) + \sum_{i=1}^{m} r_i\{0, g_i(x) - b_i\} \geq f(x^*) + \lambda^T(g(x^*) - b)$$

holds for all $x \in X_{(0,u)}$, which implies that x^* solves problem $(R_{\lambda,r,(0,z)})$. The proof for (ii) is similar to those for (i). We finish the proof. $\square$

$\boxed{3}$ Algorithms with Cuts

In this section, we present three algorithms with cuts, feasible cuts, infeasible cuts and both feasible and infeasible cuts, for problem (P) and discuss their convergence. Firstly, we state the algorithm with cuts in feasible domain.

Algorithm 3.1 (feasible cuts).

0. Select $\lambda^0,\ r^0 \in R^m_+,\ \mu > 0,\ \sigma > 0$. Set $k := 0, x^* = \emptyset, u = +\infty$.

1. Solve

$$x^{k+1} \in \arg\min_{x \in X_{(0,u)}} L(x, \lambda^k, r^k). \tag{3.1}$$

2. If

$$L(x^{k+1}, \lambda^k, r^k) \geq u, \tag{3.2}$$

Stop. Otherwise, if $g(x^{k+1}) \leq b$, set $x^* = x^{k+1}$ and $u = f(x^{k+1})$, and update λ^k and r^k by $r^{k+1} = r^k$ and $\lambda_i^{k+1} = \max\{0,\ \lambda_i^k + \mu(g_i(x^{k+1}) - b_i)\}$ for $i = 1, ..., m$. Otherwise, update λ^k and r^k by $\lambda^{k+1} = \lambda^k$ and

$$r_i^{k+1} \quad = \quad \begin{cases} r_i^k + \sigma, & g_i(x^{k+1}) - b_i > 0, \\ r_i^k, & otherwise, \end{cases} \qquad i = 1, ..., m.$$

3. Set $k = k + 1$ and go to step 1.

Remark 3.2. In order to avoid $X_{(0,u)} = \emptyset$ in Algorithm 3.1, we need to assume that there exists at least a $x \in X$ satisfying $g(x) \not\leq b$ and $f(x) < f(\hat{x})$, where $\hat{x}$ is the optimal solution of (P).

Lemma 3.3. *In Algorithm 3.1, if*

$$L(x^{k+1}, \lambda^k, r^k) \geq u$$

holds, then x^* is an optimal solution of (P).

Proof. Suppose $\hat{x}$ is an optimal solution of (P). It is clear that

$$f(\hat{x}) \leq f(x^*)$$

holds for any feasible solution x^*. Since x^{k+1} solves (3.1), we have

$$\begin{aligned} f(x^*) &\leq L(x^{k+1}, \lambda^k, r^k) \\ &\leq f(\hat{x}) + (\lambda^k)^T (g(\hat{x}) - b) \\ &\leq f(\hat{x}). \end{aligned}$$

We finish the proof. □

Theorem 3.4. *Under the assumption of Remark 3.2, Algorithm 3.1 stops at an optimal solution of (P) within finite iterations.*

Proof. If algorithm stops at k-th iteration, then from Lemma 3.3, x^* is an optimal solution. Suppose $\{x^k\}$ be the iterative sequence produced by Algorithm A. Then, there exists $N > 0$ such that for any $k > N$, x^k is infeasible for (P). Otherwise, it will be a contradiction to the facts, $\{x \in X |\ g(x) \leq b\}$ is finite and sequence $\{u^k\}$ is decreasing. For each $k > N$, all components of r^k corresponding to $\{j |\ g_j(x^k) - b_j > 0\}$ will increase σ. Thus, r^k will be large enough for sufficient large k . In other words, $f(x^*)$ is bounded and $\min\{g_j(x) - b_j > 0 |\ x \in X, g(x) \not\leq b\}$ is bounded away from zero. Therefore, there certainly exists N such that x^N satisfies (3.2). We finish the proof. □

In Algorithm 3.1, cuts act on the objective function in feasible domain. They are the decreasing upper bounds for the optimal value of (P). However, the defective of Algorithm A is the assumption in Remark 3.2. In the following, we introduce another cut which acts on the objective function on infeasible domain and no longer needs such assumption. It is a lower bound of the optimal value of problem (P).

Algorithm 3.5 (infeasible cuts).

0. Select $\lambda^0,\ r^0 \in R_+^m,\ \mu > 0,\ \sigma > 0,\ \epsilon > 0$. Set $k := 0, x^* = \emptyset, l = 0$.

1. Solve

$$x^{k+1} \in \arg \min_{x \in X_{(l,+\infty)}} L(x, \lambda^k, r^k). \tag{3.3}$$

2. If

$$L(x^{k+1}, \lambda^k, r^k) \geq f(x^*) - \epsilon, \tag{3.4}$$

stop. Otherwise, if $g(x^{k+1}) \leq b$ and $f(x^{k+1}) \leq f(x^*)$, set $x^* = x^{k+1}$; if $g(x^{k+1}) > b$, set $l = f_{r^k}(x^{k+1})$. In above two cases, update λ^{k+1}, r^{k+1} by $\lambda_i^{k+1} = \max\{0,\ \lambda_i^k + \mu(g_i(x^{k+1}) - b_i)\}$ for $i = 1, ..., m$ and $r^{k+1} = r^k$. Otherwise, update λ^{k+1} and r^{k+1} by $\lambda^{k+1} = \lambda^k$ and

$$r_i^{k+1} = \begin{cases} r_i^k + \sigma, & g_i(x^{k+1}) - b_i > 0, \\ r_i^k, & \textit{otherwise}, \end{cases} \qquad i = 1, ..., m.$$

3. Set $k = k + 1$ and go to step 1.

Before prove the convergence of Algorithm 3.5, we state the definition of ϵ-optimal solution of problem.

Definition 3.6. Suppose $\hat{x}$ is the optimal solution of (P). If x^*, a feasible solution of (P), satisfies $|f(x^*) - f(\hat{x})| \le \epsilon$, then x^* is called an ϵ-optimal solution of (P).

Lemma 3.7. *In Algorithm 3.5, if*

$$L(x^{k+1}, \lambda^k, r^k) \ge f(x^*) - \epsilon$$

holds, then x^* is an ϵ-optimal solution of (P).

Proof. The proof is similar to those for Lemma 3.3. □

Theorem 3.8. *Algorithm 3.5 would stop at the ϵ-optimal solution of problem (P) within finite iterations.*

Proof. If algorithm stop at k-th iteration, then from Lemma 3.3, x^* is an ϵ-optimal solution. Suppose $\{x^k\}$ be the iterative sequence produced by Algorithm B. Then, there exists N such that for all $k > N$, x^k is feasible for (P). The reason is that X is finite and lower cuts l^k and parameter r^k are increasing for each g-infeasible or b-infeasible point x^k. This would make the g-infeasible set be empty and the objective value at b-infeasible point be sufficiently large when k large enough. Corresponding to the feasible solutions sequence $\{x^k\}$, the multiplier sequence $\{\lambda^k\}$ is increasing and trends to zero, which means the supporting plane trends to horizon. So, there is a N such that (3.4) holds. We finish the proof. □

Now we introduce another algorithm which is the combination of Algorithm 3.1 and Algorithm 3.5.

Algorithm 3.9 (feasible and infeasible cuts).

0. Select $\lambda^0,\ r^0 \in R^m_+,\ \epsilon > 0,\ \mu > 0,\ \sigma > 0$. Set $k = 0, x^* = \emptyset, l = 0, u = +\infty$.
1. Solve
$$x^{k+1} \in \arg \min_{x \in X_{(l,u)}} L(x, \lambda^k, r^k).$$
2. If $g(x^{k+1}) \le b$, set $x^* = x^{k+1}, u = f(x^{k+1})$; if $g(x^{k+1}) > b$, set $l = f_{r^k}(x^{k+1})$. In above two cases, update λ^{k+1} and r^{k+1} by $\lambda_i^{k+1} = \max\{0,\ \lambda_i^k + \mu(g_i(x^{k+1}) - b_i)\}$ for $i = 1, ..., m$ and $r^{k+1} = r^k$. Otherwise, update λ^{k+1} and r^{k+1} by $\lambda^{k+1} = \lambda^k$ and
$$r_i^{k+1} = \begin{cases} r_i^k + \sigma, & g_i(x^{k+1}) - b_i > 0, \\ r_i^k, & otherwise, \end{cases} \qquad i = 1, ..., m.$$
3. Set $k = k + 1$ and go to step 1.

Remark 3.10. In Algorithm 3.9, we do not present stop criterion. However, we can take

$$u - l \le \epsilon, \quad L(x^{k+1}) \ge f(x^*) - \epsilon, \quad \{x \in X |\ g(x) \le b\} = \emptyset \tag{3.5}$$

as terminal criterion of Algorithm 3.9.

Theorem 3.11. *If Algorithm 3.9 takes (3.5) as terminal criterions, then it would stop at the ϵ-optimal solution of (P) within finite iterations.*

Proof. It is clear from proofs for Theorem 3.1 and Theorem 3.2. □

4 Illustrative Examples

In this section, we present some numerical examples to illustrate the algorithm process of the proposed methods. In the test, Algorithm A is used and different initial parameters for each example is chosen.

Example 4.1. As an illustrative example, let us consider Example 5.12 in Parker and Rardin (1988):

$$\begin{array}{ll} \min & f(x) = 3x_1 + 2x_2 \\ \text{s.t.} & g_1(x) = 10 - 5x_1 - 2x_2 \le 7, \\ & g_2(x) = 15 - 2x_1 - 5x_2 \le 12, \\ & x \in X = \left\{ \begin{array}{c} \text{integer} \\ 0 \le x_1 \le 1,\ 0 \le x_2 \le 2 \\ 8x_1 + 8x_2 \ge 1 \end{array} \right\}. \end{array}$$

The optimal solution of Example 4.1 is $x^* = (0,2)^T$ and the corresponding objective function is $f^* = 4$. The iterative results of Algorithm 3.1, 3.5 and 3.9 for given $\mu = \sigma = 1$ and four different kinds of λ^0 and r^0 are listed in table 1. In the table, ITER denotes the number of iterations and u, l the final values of u, l at algorithm termination.

Table 1

Algorithm	$(\lambda^0)^T$	$(r^0)^T$	x^*	u	l	ITER
A	(1,1)	(1,1)	(0,2)	4		5
B	(1,1)	(1,1)	(0,2)		0	5
C	(1,1)	(1,1)	(0,2)	4	0	5
A	(10,10)	(10,10)	(0,2)	4		4
B	(10,10)	(10,10)	(0,2)		0	4
C	(10,10)	(10,10)	(0,2)	0	0	4
A	(50,50)	(100,100)	(0,2)	4		4
B	(50,50)	(100,100)	(0,2)		0	4
C	(50,50)	(100,100)	(0,2)	4	0	4
A	(50,50)	(50,50)	(0,2)	4		9
B	(50,50)	(50,50)	(0,2)		0	9
C	(50,50)	(50,50)	(0,2)	4	0	9

Example 4.2.

$$\begin{array}{ll} \min & f(x) = 33 - (5x_1 + 9x_2 + 2x_3 + 7x_4 + 4x_5 + 6x_6) \\ \text{s.t.} & g_1(x) = 2x_1 + 7x_2 - 2x_3 + 4x_4 - x_5 + 6x_6 \le 12 \\ & g_2(x) = 2x_1 + 6x_2 + 2x_3 + 5x_4 + 5x_5 \le 14 \\ & g_3(x) = 3x_1 + 3x_2 - x_3 + 2x_4 + 5x_6 \le 8 \\ & x \in X = \{x | x_j = 0, 1; j = 1, 2, ..., 6\} \end{array}$$

The optimal solution of Example 4.2 is $x^* = (0,1,1,0,1,1)^T$ and the corresponding objective function is $y^* = 12$. The iterative results of Algorithm 3.1, 3.5 and 3.9 for parameters $\mu = \sigma = 1$ and three different kinds of λ^0 and r^0 are listed in table 2 as follows. In the table, ITER denotes the number of iterations, and u, l the final values of u, l at algorithm termination.

Table 2

Algorithm	$(\lambda^0)^T$	$(r^0)^T$	x^*	u	l	ITER
A	(1,1,1)	(1,1,1)	(0,1,1,0,1,1)	12		6
B	(1,1,1)	(1,1,1)	(0,1,1,0,1,1)		0	6
C	(1,1,1)	(1,1,1)	(0,1,1,0,1,1)	12	0	6
A	(50,50,50)	(1,1,1)	(0,1,1,0,1,1)	12		13
B	(50,50,50)	(1,1,1)	(0,1,1,0,1,1)		0	13
C	(50,50,50)	(1,1,1)	(0,1,1,0,1,1)	12	0	13
A	(100,100,100)	(100,100,100)	(0,1,1,0,1,1)	12		15
B	(100,100,100)	(100,100,100)	(0,1,1,0,1,1)		0	15
C	(100,100,100)	(100,100,100)	(0,1,1,0,1,1)	12	0	15

Example 4.3.

$$\begin{array}{ll} \min & f(x) = 60 - 2x1 - 3x2 - 10x3 \\ \text{s.t.} & g_1(x) = -x_1 + 4x_2 + 7x_3 \leq 10 \\ & g_2(x) = 5x_1 - 2x_2 + 6x_3 \leq 10 \\ & x \in X = \{x | 0 \leq x_j \leq 4, integer, j = 1,2,3\} \end{array}$$

The optimal solutions of Example 4.3 are $x* = (1,1,1)^T$, $(3,3,0)^T$ and optimal value is $y^* = 45$. The testing results of Algorithm 3.1, 3.5 and 3.9 for parameters $\mu = \sigma = 1$ and three different kinds of λ^0 and r^0 are listed in table 3. In the table, ITER denotes the number of iterations, and u, l the final values of u, l at algorithm termination.

Table 3

Algorithm	$(\lambda^0)^T$	$(r^0)^T$	x^*	u	l	ITER
A	(10,10)	(10,10)	(1,1,1)	45		3
B	(10,10)	(10,10)	(1,1,1)		0	3
C	(10,10)	(10,10)	(1,1,1)	45	0	3
A	(25,25)	(10,10)	(1,1,1)	45		6
B	(25,25)	(10,10)	(1,1,1)		0	6
C	(25,25)	(10,10)	(1,1,1)	45	0	6
A	(50,50)	(50,50)	(3,3,0)	45		24
B	(50,50)	(50,50)	(3,3,0)		0	24
C	(50,50)	(50,50)	(3,3,0)	45	0	24

As witnessed from the above three examples, the proceeding of objective cuts provide a valuable way which transfers an integer programming problem with multiple constraints into an augmented Lagrangian problem. We also observe that how to choose the initial parameters and how to adjust parameters in iterations for different problems is infective to the rate of convergence.

Acknowledgement

The authors would like to thank the anonymous referees for providing many invaluable comments and suggestions that lead to significant improvement of this paper.

References

[1] G. Cohen and Z.L. Zhu, Decomposition coordination methods in large scale optimization problems: the nondifferential case and the use of augmented Lagrangian, *Advances in Large Scale System* 1 (1984) 203–266.

[2] M.L. Fisher, The Lagrangian relaxation method for solving integer programming problems, *Management Sci.* 27 (1981) 1–18.

[3] A.M. Geoffrion, Lagrangian relaxation for integer programming, *Math. Programming Stud.* 2 (1974) 82–114.

[4] M. Guignard and S. Kim, Lagrangian decomposition: a model yielding stronger Lagrangian relaxation bounds, *Math. Program.* 33 (1993) 262–273.

[5] D. Li, Zero duality gap in integer programming: P-norm surrogate constraint method, *Oper. Res. Lett.* 25 (1999) 89–96.

[6] D. Li, J. Wang and X.L. Sun, Computing exact solution to nonlinear integer programming: Convergent Lagrangian and objective level cut method, *J. Global Optim.* 39 (2007) 127–154.

[7] D. Li and X.L. Sun, Success guarantee of dual search in integer programming:p-th norm power Lagrangian method, *J. Global Optim.* 18 (2000) 235–254.

[8] D.C. Llewellyn and J. Ryan, A primal dual integer programming algorithm, *Discrete Applied mathematics* 45 (1993) 262–269.

[9] Michelon, P. and Maculan, N., Lagrangian methods for $0-1$ quadratic programming, *Discrete Applied Mathematics* 42 (1993) 257–269.

[10] P. Michelon and N. Maculan, Lagrangian decomposition for integer nonlinear programming with linear constraints, *Math. Program.* 52 (1991) 303–313.

[11] R.G. Parker and R.L. Rardin, *Discrete Optimization*, Academic, Boston, 1988.

[12] J.F. Shapiro, A survey of Lagrangian techniques for discrete optimization, *Annals of Discrete Mathematics* 5 (1979) 113–138.

[13] X.L. Sun and D. Li, Asympototic strong duality for bounded integer programming: a logarithmic-exponential dual formulation, *Math. Oper. Res.* 25 (2000) 625–644.

[14] H.P. Williams, Duality in mathematics and linear and integer programming, *J. Optim. Theory Appl.* 90 (1996) 257–278.

Manuscript received 12 June 2007
revised 31 August 2007
accepted for publication 3 September 2007

Wei Wang
Department of Applied Mathematics, East China University of Science and Tech.
Shanghai, 200237, China
E-mail address: wangwei@ecust.edu.cn

Yifan Xu
School of Management, Fudan University
Shanghai, 200433, China
E-mail address: yfxu@fudan.edu.cn