Electronic Colloquium on Computational Complexity, Report No. 67 (2007)

Computing 1/3-approximate Nash equilibria of bimatrix games in polynomial time [*]

Haralambos Tsaknakis [†]   Paul G. Spirakis [†] [‡]

May 22, 2007

Abstract

In this paper we propose a methodology for determining approximate Nash equilibria of non-cooperative bimatrix games and, based on that, we provide a polynomial time algorithm that computes $\frac{1}{3} + \frac{1}{p(n)}$ -approximate equilibria , where $p(n)$ is a polynomial controlled by our algorithm and proportional to its running time. The methodology is based on the formulation of an appropriate function of pairs of mixed strategies reflecting the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other. We then seek to minimize such a function using descent procedures. As it is unlikely to be able to find global minima in polynomial time, given the recently proven intractability of the problem, we concentrate on the computation of local minima and prove that they can be approximated arbitrarily close in polynomial time and that they have the above mentioned approximation property. Our result provides the best ϵ till now for polynomially computable ϵ-approximate Nash equilibria of bimatrix games.

1 Introduction

Ever since it was proved that the problem of finding exact Nash equilibria is intractable in the sense that it is PPAD-complete even for 2-player games [1], attention has been focused on finding ϵ-approximate such equilibria for $\epsilon > 0$. In this respect, simple algorithms have recently been provided for finding approximate equilibria for constant $\epsilon = \frac{3}{4}$ and $\epsilon = \frac{1}{2}$ ([3] , [4]) for general bimatrix games (and for positively normalised payoff matrices) based on examining small supports of 1 or 2 for either player. The best known result so far provides 0.38-approximate Nash equilibria of normalised bimatrix games in polynomial time ([2]). Furthermore , it has been shown ([5]) that the more general approximation problem of finding a fully polynomial time approximation scheme for any $\epsilon > 0$, has similar complexity with the problem of finding exact Nash equilibria.

[*]This work has been partially supported by the IST Programme of the European Union DELIS

[†]Research Academic Computer Technology Institute, RACTI , Greece , Email: {tsaknak,spirakis}@cti.gr

[‡]Dept. of Computer Eng. and Informatics , Patras Univ. , Patras Greece

ISSN 1433-8092

For a different, stronger, notion of approximation, i.e. the well supported approximate Nash equilibria, the best known result so far provides 0.658-approximate well supported equilibria for normalised bimatrix games in polynomial time ([6]).

Most of the reported investigations of finding approximate equilibria for constant ϵ are based on the examination of small supports of the strategy sets of the players and the algorithms presented are based on brute force search over all such supports.

In this work we adopt a different approach that does not rely on any pre-specified small supports neither on an indiscriminate search over all small support strategies. We define an equivalent optimization problem in the strategy spaces of both players and attempt to obtain a local optimum of a specific function that measures the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other. We prove that at any local minimum of that function we obtain a strategy pair that is an $\frac{1}{3}$-approximate Nash equilibrium. We also prove that an almost local minimum of the function can be reached in polynomial time with respect to the input data of the game , and that point suffices to get arbitrarily close to $\frac{1}{3}$.

2 Definitions and notation

Let R, C denote the m by n row and column players' payoff matrices respectively , for m, n any positive integers. We assume that both payoff matrices are positively normalized , i.e. all their entries belong to $[0, 1]$ (without loss of generality any game can be equivalently transformed to a positively normalized game by appropriate shifting and scaling each one of the payoff matrices).

Let us denote by e_k the k-dimensional column vector having all its entries equal to 1 (for positive integer k). Let

$$\Delta_k = \{u : u \in R^k, u \geq 0, e_k^\tau u = 1\}$$

be the k-dimentional standard simplex (superscript τ denotes transpose).

Also , for any vector $u \in R^k$, we define the following :

$$supp(u) = \{i \in (1, k) : u_i \neq 0\}$$

being the support index subset of $u \in R^k$ and also

$$suppmax(u) = \{i \in (1, k) : u_i \geq u_j \ \forall j \in (1, k)\}$$

being the index subset where all entries are equal to the maximum entry of $u \in R^k$.

We also denote by

$$max(u) = \{u_i : u_i \geq u_j, \text{for all } j\} \tag{1}$$

the value of the maximum entry of the vector.

The problem of finding an ϵ-approximate Nash equilibrium in the game (R, C) , for some $\epsilon \geq 0$, is to compute a pair of strategies $\overline{x}$ in Δ_m and $\overline{y}$ in Δ_n such that the following relationships hold :

$$x^{\tau} R\overline{y} \leq \overline{x}^{\tau} R\overline{y} + \epsilon \text{ for all } x \in \Delta_m$$

and

$$\overline{x}^{\tau} C y \leq \overline{x}^{\tau} C\overline{y} + \epsilon \text{ for all } y \in \Delta_n$$

3 Optimization formulation

Key to our approach is the definition of the following function mapping $\Delta_m \times \Delta_n$ into $[0, 1]$:

$$f(x, y) = max\{max(Ry) - x^{\tau} Ry, max(C^{\tau} x) - x^{\tau} Cy\}$$

It is evident that $f(x, y) \geq 0 \; \forall \; (x, y) \in \Delta_m \times \Delta_n$ and that exact Nash equilibria of (R, C) correspond to pairs of strategies such that $f(x, y) = 0$. Furthermore, ϵ- approximate equilibria correspond to strategy pairs that satisfy $f(x, y) \leq \epsilon$. This function represents the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other.

The function $f(x, y)$ is not jointly convex with respect to both x and y . However , it is convex in x alone , if y is kept fixed and vice versa.

Let us define the two ingredients of the function $f(x, y)$ as follows :

$f_R(x, y) = max(Ry) - x^{\tau} Ry$

and

$f_C(x, y) = max(C^{\tau} x) - x^{\tau} Cy$

From any point in $(x, y) \in \Delta_m \times \Delta_n$ we consider variations of $f(x, y)$ along feasible directions in both players' strategy spaces of the following form :

$$(1 - \epsilon) \begin{bmatrix} x \\ y \end{bmatrix} + \epsilon \begin{bmatrix} x' \\ y' \end{bmatrix}$$

where , $0 \leq \epsilon \leq 1, (x', y') \in \Delta_m \times \Delta_n$

(the vectors in brackets are $m + n$ - dimensional column vectors).

The variation of the function along such a feasible direction is given by the following relationship :

$$Df(x, y, x', y', \epsilon) = f(x + \epsilon(x' - x), y + \epsilon(y' - y)) - f(x, y)$$

Considering sufficiently small values of ϵ (that will be examined later) and ignoring second order terms in the above relationship , the gradient at the point (x, y) along an arbitrary direction specified by another point (x', y') has been derived as follows :

$$Df(x, y, x', y') = lim(for \; \epsilon \rightarrow 0) \; of \; ((\frac{1}{\epsilon}) Df(x, y, x', y', \epsilon)$$

Let us now define :

$$S_R(y) = suppmax(Ry) \; and \; S_C(x) = suppmax(C^\tau x)$$

From the Appendix we get :
(a) If $f_R(x,y) = f_C(x,y)$ then

$$Df(x,y,x',y') = max(T_1(x,y,x',y'), T_2(x,y,x',y')) - f(x,y)$$

where

$$m_1(y') = max(Ry') \; over \; the \; subset \; S_R(y)$$

and

$$m_2(x') = max(C^\tau x') \; over \; the \; subset \; S_C(x)$$

and

$$T_1(x,y,x',y') = m_1(y') - x^\tau Ry' - (x')^\tau Ry + x^\tau Ry$$

and

$$T_2(x,y,x',y') = m_2(x') - x^\tau Ry' - (x\prime)^\tau Cy + x^\tau Cy$$

(b) If $f_R(x,y) > f_C(x,y)$ then
$Df(x,y,x',y') = T_1(x,y,x',y') - f(x,y)$
and
(c) If $f_R(x,y) < f_C(x,y)$ then
$Df(x,y,x',y') = T_2(x,y,x',y') - f(x,y)$. In the cases (b) and (c) the functions T_1 and T_2 are as defined in case (a).

The problem of finding $Df(x,y)$ as the minimum over all $(x',y') \in \Delta_m \; by \; \Delta_n$ of the function $Df(x,y,x',y')$, is a linear programming problem.

This problem can be equivalently expressed as the following mini-max problem by introducing appropriate dual variables (we derive it for (x,y) such that $f_R(x,y) = f_C(x,y)$ since the cases where the two terms are different can be reduced to this by solving an LP , as we shall see below) :

Minimize (over x', y') the maximum (over w, z, ρ) of the function

$$[\rho w^\tau, (1-\rho)z^\tau] G(x,y) \begin{bmatrix} y' \\ x' \end{bmatrix}$$

where :
(a) the maximum is taken with respect to dual variables w, z, ρ such that :
$w \in \Delta_m, supp(w) \subset S_R(y)$ and $z \in \Delta_n, supp(z) \subset S_C(x)$ and $\rho \in [0,1]$.
(b) The minimum is taken with respect to $(x',y') \in \Delta_m \; by \; \Delta_n$, and
(c) the matrix $G(x,y)$ is the following $(m+n) \; by \; (m+n)$ matrix :

$$G(x,y) = \left[\begin{array}{cc} R - e_m x^\tau R & -e_m y^\tau R^\tau + e_m {e_m}^\tau x^\tau R y \\ -e_n x^\tau C + e_n {e_n}^\tau x^\tau C y & C^\tau - e_n y^\tau C^\tau \end{array} \right]$$

Solving the above mini-max problem we obtain w, z, ρ, x' and y' that are all functions of the point (x, y) and take values in their respective domains of definition. Let us denote by $V(x, y)$ the value of the solution of the mini-max problem at the point (x, y) . The solution of this problem yields a feasible descent direction (as a matter of fact the steepest feasible descent direction) for the function $f(x, y)$ if $Df(x, y) = V(x, y) - f(x, y) < 0$. Following such a descent direction we can perform an appropriate line search with respect to the parameter ϵ and find a new point that gives a lower value of the function $f(x, y)$. Applying repeatedly such a descent procedure we will eventually reach a point where no further reduction is possible. Such a point is a local minimum (or stationary point) that satisfies $Df(x, y) \geq 0$.

In order to control the number of repetitions of the descent procedure , we do the following (without loss of generality , let us assume that $n \geq m$) :

Procedure descent (p (n))

(Note : here $p(n)$ is a suitably chosen polynomial in n , e.g. $p(n) = n^\alpha, \alpha > 1$) .

1. Start with an arbitrary $(x, y) = (x_0, y_0)$ in $\Delta_m \times \Delta_n$ (e.g. the uniform distribution). Produce another pair (x, y) with lower value of $f(x, y)$ and for which $f_R(x, y) = f_C(x, y)$ as follows :

(a) If $f_R(x_0, y_0) > f_C(x_0, y_0)$, keep y_0 fixed and solve the LP :
minimize (over $x \in \Delta_m$) the

$$max(Ry_0) - x^\tau R y_0$$

under the constraint :

$$max(C^\tau x) - x^\tau C y_0 \leq max(Ry_0) - x^\tau R y_0$$

(b) If $f_R(x_0, y_0) < f_C(x_0, y_0)$, keep x_0 fixed and solve the LP :
minimize (over $y \in \Delta_n$) the

$$max(C^\tau x_0) - {x_0}^\tau C y$$

under the constraint :

$$max(Ry) - {x_0}^\tau R y \leq max(C^\tau x_0) - {x_0}^\tau R y$$

2. Compute the value $V(x, y)$ and the corresponding (x', y') by solving the linear mini-max problem (with the matrix $G(x, y)$ as defined above).

3. If $V(x, y) - f(x, y) < 0$ but $V(x, y) - f(x, y) > -\frac{1}{p(n)}$ stop (almost local minimum) and exit.

Also , if $V(x, y - f(x, y) \geq 0$, then stop (local minimum) and exit.

4. If the conditions in step 3 are not satisfied , compute the minimum with respect to ϵ of the function $f(x + \epsilon(x' - x), y + \epsilon(y' - y))$ along the direction in the (x, y) space

specified by the (x', y') found in step 2 , until the first breakpoint of the variable ϵ , and set $(x, y) = (x + \epsilon(x' - x), y + \epsilon(y' - y))$. (Notice that the number of breakpoints for ϵ one needs to consider for finding the global minimum of $f(x + \epsilon(x' - x), y + \epsilon(y' - y))$ with respect to ϵ is of the order of $O(n)$ (see the formula in the Appendix)). Furthermore , if for the new (x, y) we have $f_R(x, y) \neq f_C(x, y)$, solve the LP specified in step 1 and compute the new (x, y) with lower value of the function $f(x, y)$ and for which $f_R(x, y) = f_C(x, y)$.

Go to step 2 .

end of descent .

4 Approximation properties of local minima

Let us assume that we have a local minimum $(x^\star, y^\star)$ of the function $f(x, y)$. Then , based on the above analysis and notation , the following relationship should be true :

$$Df(x^\star, y^\star) = V(x^\star, y^\star) - f(x^\star, y^\star) \geq 0$$

Let $(w^\star, z^\star) \in \Delta_m \times \Delta_n, \rho^\star \in [0, 1]$ be a solution of the linear mini-max problem (with matrix $G(x^\star, y^\star)$) with respect to the dual variables corresponding to the pair $(x^\star, y^\star)$. Such a solution should satisfy the relations $supp(w^\star) \subset S_R(y^\star)$ and $supp(z^\star) \subset S_R(x^\star)$.

Let us define a pair of strategies $(\hat{x}, \hat{y}) \in \Delta_m \times \Delta_n$ as follows :

$$(\hat{x}, \hat{y}) = \left\{ \begin{array}{ll} (x^\star, y^\star), & \text{if } f(x^\star, y^\star) \leq f(w^\star, z^\star) \\ (w^\star, z^\star) & \text{, otherwise} \end{array} \right\}$$

We now express the main result of this paper in the following theorem :

Theorem 1 *The pair of strategies $(\hat{x}, \hat{y})$ defined above , is a $\frac{1}{3}$ - approximate Nash equilibrium.*

proof

From the definition of $(\hat{x}, \hat{y})$ we have :

$$f(\hat{x}, \hat{y}) \leq min\{f(x^\star, y^\star), f(w^\star, z^\star)\} \tag{2}$$

Using the stationarity condition for $(x^\star, y^\star)$ we obtain :

$$f(x^\star, y^\star) \leq V(x^\star, y^\star)$$

But $V(x^\star, y^\star)$ is less than or equal to

$$\rho^\star E_1 + (1 - \rho^\star)E_2$$

where

$$E_1 = (w^{\star\tau} R y' - x^{\star\tau} R y' - x'^{\tau} R y^\star + x^{\star\tau} R y^\star)$$

and

$$E_2 = (z^{\star\tau} C^{\tau} x' - x^{\star\tau} C y' - x'^{\tau} C y^{\star} + x^{\star\tau} C y^{\star})$$

and this holds $\forall (x', y') \in \Delta_m \times \Delta_n$
Setting $x' = x^{\star}$ and $y' : supp(y') \subset S_C(x^{\star})$ in the above inequality we get :

$$f(x^{\star}, y^{\star}) \leq \rho^{\star} \lambda \tag{3}$$

where λ is the minimum over all $y' : supp(y') \subset S_C(x^{\star})$ of the quantity :

$$\{(w^{\star} - x^{\star})^{\tau} R y'\}$$

Next , setting $y' = y^{\star}$ and $x' : supp(x') \subset S_R(y^{\star})$ in the same inequality , we get :

$$f(x^{\star}, y^{\star}) \leq (1 - \rho^{\star}) \mu \tag{4}$$

where μ is the minimum over all $x' : supp(x') \subset S_R(y^{\star})$ of the quantity :

$$\{x'^{\tau} C (z^{\star} - y^{\star})\}$$

Now , since $supp(z^{\star}) \subset S_C(x^{\star})$, we have $(w^{\star} - x^{\star})^{\tau} R z^{\star} \geq \lambda$ which implies :

$$max(R z^{\star}) - w^{\star\tau} R z^{\star} \leq max(R z^{\star}) - x^{\star\tau} R z^{\star} - \lambda \leq 1 - \lambda$$

Similarly , since $supp(w^{\star}) \subset S_R(y^{\star})$, we have $w^{\star\tau} C (z^{\star} - y^{\star}) \geq \mu$ which implies :

$$max(C^{\tau} w^{\star}) - w^{\star\tau} C z^{\star} \leq max(C^{\tau} w^{\star}) - w^{\star\tau} C y^{\star} - \mu \leq 1 - \mu$$

From the last two inequalities we deduce :

$$f(w^{\star}, z^{\star}) \leq 1 - min(\lambda, \mu) \tag{5}$$

Combining inequalities (3) , (4) , (5) and the definition of $(\hat{x}, \hat{y})$, we get :

$$f(\hat{x}, \hat{y}) \leq min\{\rho^{\star}\lambda, (1 - \rho^{\star})\mu, 1 - min(\lambda, \mu)\} \tag{6}$$

By straightforward calculation , the right hand side of the above inequality (6) cannot exceed the number $\frac{1}{3}$ for any $\rho^{\star}, \lambda, \mu \in [0, 1]$, which proves our claim that $(\hat{x}, \hat{y})$ is $\frac{1}{3}$ - approximate Nash equilibrium.

This concludes the proof of our main Theorem .

Now , if we have an almost local minimum $({x_d}^{\star}, {y_d}^{\star})$ instead of an exact one , the stationarity condition (used in the proof of the theorem) becomes :

$$f({x_d}^{\star}, {y_d}^{\star}) \leq V({x_d}^{\star}, {y_d}^{\star}) + \frac{1}{p(n)}$$

Following the same steps of the proof as above , with all quantities determined by such a point $({x_d}^{\star}, {y_d}^{\star})$, the inequalities (3) and (4) are modified as :

$$f(x_d{}^\star, y_d{}^\star) \leq \rho_d{}^\star \lambda_d + \frac{1}{p(n)}$$

and

$$f(x_d{}^\star, y_d{}^\star) \leq (1 - \rho_d{}^\star) \mu_d + \frac{1}{p(n)}$$

Also , inequality (5) remains the same with respect to the quantities λ_d, μ_d . Therefore , using the inequality (6) with these bounds we end up with a $\frac{1}{3} + \frac{1}{p(n)}$ - approximate equilibrium.

5 The complexity of our algorithm

Our algorithm is basically the procedure descent of the function $f(x, y)$. This takes at most $p(n) T_L P(n)$ time to run (when $n \geq m$) where $T_L P(n)$ is the time to solve a linear program of size n.

An arbitrary point $(x, y) \in \Delta_m \times \Delta_n$ can be used to initialize the algorithm.

6 Discussion and future work

It is known from Bellare and Rogaway ([7]) that (even in a weaker sense) there is no polynomial time μ - approximation of the optimal value of the problem $min\{x^\tau Qx, s.t. Bx = b, 0 \leq x \leq e\}$ for some $\mu \in (0, \frac{1}{3})$, unless $P = NP$. Of course , here μ is a multiplicative relative accuracy and the reduction that they use involves matrices that are different from the ones in our case. However , this gives evidence that going below $\frac{1}{3}$ in the approximation of equilibria will probably require a radically different approach (if any) , perhaps probabilistic. We are currently working on this.

References

[1] X. Chen , X. Deng . Settling the complexity of 2-player Nash equilibrium . In Proc. of the 47th IEEE Symp. on Foundations of Comp. Sci. (FOCS 06) , pp. 261-272 , 2006.

[2] C. Daskalakis , A. Mehta , C. Papadimitriou . Progress in approximate Nash Equilibria . In Proc. of the 8th ACM Conf. on Electronic Commerce (EC '07) , 2007 (to appear).

[3] C. Daskalakis , A. Mehta , C. Papadimitriou . A note on approximate Nash equilibria . In Proc. of the 2nd workshop on the Internet and Network Economics (WINE '06) , vol. 4286 LNCS , pp. 297-306 , Springer , 2006.

[4] S. Kontogiannis , P. Panagopoulou , P. G. Spirakis Polynomial algorithms for approximating Nash equilibria in bimatrix games . In Proc. of the 2nd workshop on the Internet and Network Economics (WINE 2006) , vol. 4286 LNCS , pp. 282-296 , Springer , 2006.

[5] X. Chen , X. Deng and S. Teng . Computing Nash Equilibria : Approximation and smoothed complexity. In Proc. of the 47th IEEE Symp. on Foundations of Comp. Sci. (FOCS 06) , pp. 603-612 , IEEE Press , 2006.

[6] S. Kontogiannis , P. G. Spirakis . Efficient algorithms for constant well supported approximate equilibria of bimatrix games . ICALP 2007 (to appear).

[7] M. Bellare , P. Rogaway . The complexity of approximating a nonlinear program. Mathematical Programming , 69:429-441 , 1995.

A Appendix

Using the definitions for any $(x, y) \in \Delta_m \times \Delta_n$ i.e :

$$\begin{aligned} f_R(x,y) &= max(Ry) - x^\tau Ry \\ f_C(x,y) &= max(C^\tau x) - x^\tau Cy \\ f(x,y) &= max\{f_R(x,y), f_C(x,y)\} \end{aligned}$$

we have , for any $(x', y') \in \Delta_m \times \Delta_n$ and any $\epsilon \in [0, 1]$ that :

$$Df(x, y, x', y', \epsilon) = f(x + \epsilon(x' - x), y + \epsilon(y' - y)) - f(x, y)$$

This can be written as (analytically)

$$max\{f_R(x+\epsilon(x'-x), y+\epsilon(y'-y)), f_C(x+\epsilon(x'-x), y+\epsilon(y'-y))\} - max\{f_R(x,y), f_C(x,y)\}$$

and this is actually $max(K_1, K_2)$ where

$$K_1 = \epsilon Df_R + \Lambda f_R - \epsilon^2 Hf_R - (1-\epsilon)max\{0, f_C(x,y) - f_R(x,y)\}$$

and also

$$K_2 = \epsilon Df_C + \Lambda f_C - \epsilon^2 Hf_C - (1-\epsilon)max\{0, f_R(x,y) - f_C(x,y)\}$$

where now the functions $Df_R, \Lambda f_R, Hf_R, Df_C, \Lambda f_C, Hf_C$ are defined below.

$$\begin{aligned} Df_R(x,y,x',y') = \{max(Ry')over S_R(y)\} - x^\tau Ry' - x'^\tau Ry + x^\tau Ry - f(x,y) \\ and \\ Hf_R(x,y,x',y') = (x'-x)^\tau R(y'-y) \\ and \\ Df_C(x,y,x',y') = \{max(C^\tau x')over S_C(x)\} - x^\tau Cy' - x'^\tau Cy + x^\tau Cy - f(x,y) \\ and \\ Hf_C(x,y,x',y') = (x'-x)^\tau C(y'-y) \end{aligned}$$

In order to define $\Lambda f_R, \Lambda f_C$ we remind the reader that $S_R(y) = suppmax(Ry)$ and that $S_C(x) = suppmax(C^\tau x)$ and we will also use their complements :

$\bar{S_R}(y)$ being the complement of $S_R(y)$ in the index set $\{1, m\}$ and

$\bar{S_C}(x)$ being the complement of $S_C(x)$ in the index set $\{1, n\}$
Let now

M_y be the maximum of Ry over $S_R(y)$

$M_{y'}$ be the maximum of Ry' over $S_R(y)$

and

M_x be the maximum of $C^\tau x$ over $S_C(x)$

$M_{x'}$ be the maximum of $C^\tau x'$ over $S_C(x)$

Finally $\Lambda f_R(x, y, x', y', \epsilon)$ is the maximum of
(0 , max over $\bar{S_R}(y)$ of $(I(y, y') + J(y)$)) where

$I(y, y') = \epsilon((Ry' - e_m M_{y'}) + (M_y e_m - Ry))$ and

$J(y) = -(M_y e_m - Ry)$

Also finally $\Lambda f_C(x, y, x', y', \epsilon)$ is also the maximum of
((0 , max over $\bar{S_C}(x)$ of $(I(x, x') + J(x)$)) where

$I(x, x') = \epsilon((C^\tau x' - e_n M_{x'}) + (M_x e_n - C^\tau x))$ and

$J(x) = -(M_x e_n - C^\tau x)$

From the above equations , the gradient at the point $(x, y) \in \Delta_m \times \Delta_n$ along a feasible direction specified by a $(x', y') \in \Delta_m \times \Delta_n$ can be determined by letting ϵ go to 0 and get finally :

$$Df(x, y, x', y') = \left\{ \begin{array}{ll} max(Df_R, Df_C) & \text{if } f_R(x, y) = f_C(x, y) \\ Df_R & \text{if } f_R(x, y) > f_C(x, y) \\ Df_c & \text{if } f_R(x, y) < f_C(x, y) \end{array} \right\}$$