Decoding Downset Codes over Finite Grids

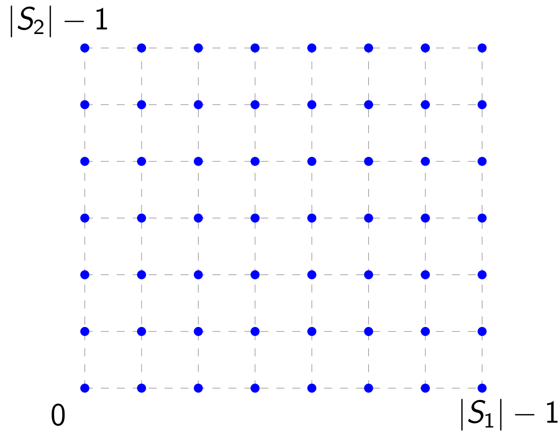
S. Venkitesh
Department of Mathematics, IIT Bombay

Diamond Jubilee Symposium
Department of Mathematics, IIT Bombay
January 4, 2019

Introduction

Field $\quad \mathbb{F} \qquad$ (arbitrary)

Finite grid $\qquad$ (nonempty, finite set) $\qquad S = S_1 \times \cdots \times S_n = \{p_1, \ldots, p_k\} \subseteq \mathbb{F}^n$



$$f : S \to \mathbb{F}$$

$$\updownarrow$$

$$f = (f(p_1), \ldots, f(p_k))$$

$$\updownarrow$$

$$f = P : S \to \mathbb{F}$$

(polynomial function)

$$\deg_{X_i} P(\mathbf{X}) \leq |S_i| - 1, \ \forall\, i \in [n]$$

$$\mathcal{M}_S = \{\mathbf{X}^\alpha = X_1^{\alpha_1} \cdots X_n^{\alpha_n} : \alpha_i \leq |S_i| - 1, \ \forall\, i \in [n]\}$$

$$\{S \to \mathbb{F}\} \simeq \mathrm{span}_{\mathbb{F}} \mathcal{M}_S$$

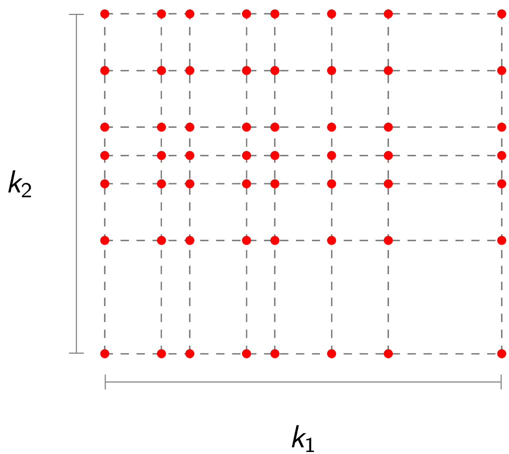
$$\{S \to \mathbb{F}\} = \mathrm{span}_{\mathbb{F}} \mathcal{M}_S$$

Introduction

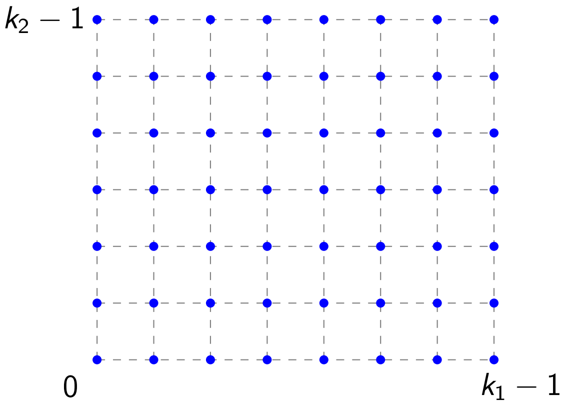
$$S = S_1 \times \cdots \times S_n \subseteq \mathbb{F}^n, \quad k_i = |S_i|, \forall\, i \in [n], \quad k = |S| = k_1 \cdots k_n$$

$$\mathcal{M}_S = \{\mathbf{X}^\alpha = X_1^{\alpha_1} \cdots X_n^{\alpha_n} : \alpha_i \leq k_i - 1, \forall\, i \in [n]\} = \mathbb{F}\text{-basis of } \{S \to \mathbb{F}\}$$

$S \subseteq \mathbb{F}^2$



$\mathcal{M}_S \subseteq \mathbb{N}^2$



Introduction

Definition (Reed-Muller Code on S)

$$RM(S,d) = \{P(\mathbf{X}) \in \mathrm{span}_{\mathbb{F}}\mathcal{M}_S \mid \deg P \leq d\}, \quad d \leq \sum_{i=1}^{n}(k_i - 1)$$

$RM(S,d)$ is a subspace of $\mathrm{span}_{\mathbb{F}}\mathcal{M}_S \simeq \mathbb{F}^k$, i.e. it is a **linear code**.

$$RM(S,d) = \mathrm{span}_{\mathbb{F}}\mathcal{M}_{S,d}, \quad \text{where } \mathcal{M}_{S,d} = \left\{\mathbf{X}^\alpha \in \mathcal{M}_S : |\alpha| = \sum_{i=1}^{n}\alpha_i \leq d\right\}.$$

Note that

$$\mathbf{X}^\alpha \in \mathcal{M}_{S,d}, \mathbf{X}^\beta \mid \mathbf{X}^\alpha \text{ (i.e. } \beta \leq \alpha \text{ in natural partial order)} \quad \Longrightarrow \quad \mathbf{X}^\beta \in \mathcal{M}_{S,d}.$$

So $\mathcal{M}_{S,d}$ is a **down-closed set (downset)**.

Downset and Downset Code

Definition (Downset)

A nonempty set of monomials $\mathcal{D}$ is called a **downset** if

$$\mathbf{X}^{\alpha} \in \mathcal{D},\ \ \mathbf{X}^{\beta} \mid \mathbf{X}^{\alpha}\ \ (\beta \leq \alpha) \quad \Longrightarrow \quad \mathbf{X}^{\beta} \in \mathcal{D}.$$

Definition (Downset code on S)

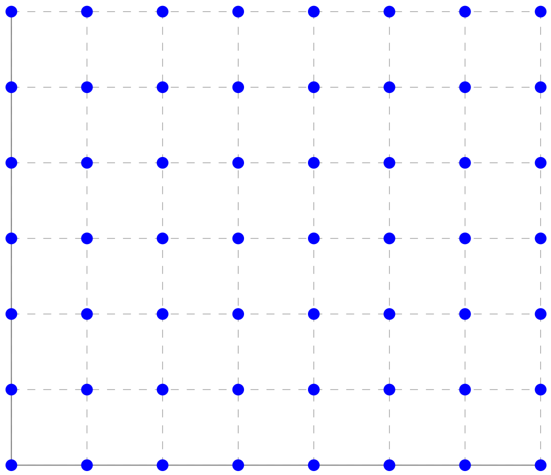
$$\mathcal{C}(S, \mathcal{D}) = \mathrm{span}_{\mathbb{F}} \mathcal{D}, \quad \text{where } \mathcal{D} \subseteq \mathcal{M}_S \text{ is a downset}$$

$\mathcal{C}(S, \mathcal{D})$ is a subspace of $\mathrm{span}_{\mathbb{F}} \mathcal{M}_S \simeq \mathbb{F}^k$, i.e., it is a linear code.

The downset $\mathcal{D}$ is an $\mathbb{F}$-basis of $\mathcal{C}(S, \mathcal{D})$.

Examples of Downsets and Downset codes

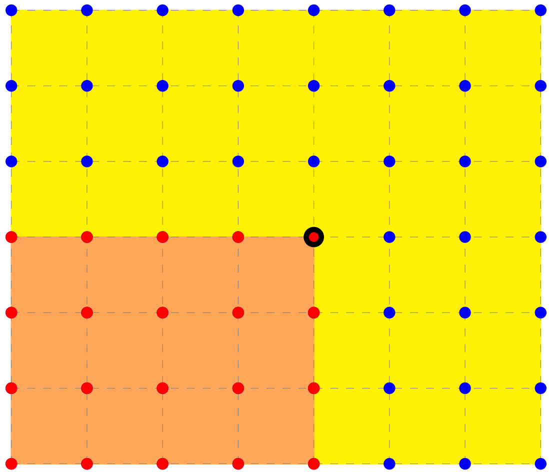
Eg. $n = 2$, $S = S_1 \times S_2$, $|S_1| = 8$, $|S_2| = 7$, $\mathcal{M}_S = \{0, \ldots, 7\} \times \{0, \ldots, 6\}$



$\mathcal{M}_S$

Examples of Downsets and Downset codes

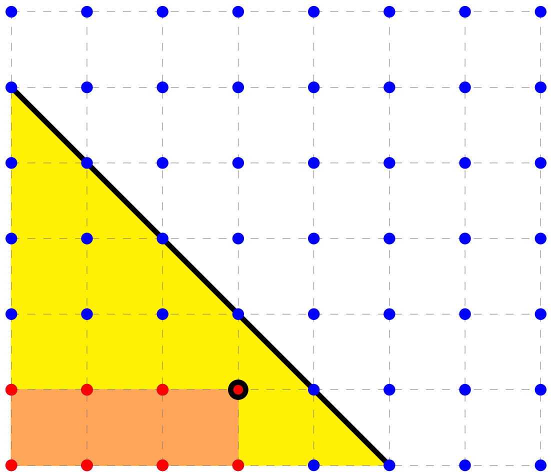
Eg. $n = 2, \quad S = S_1 \times S_2,\ |S_1| = 8,\ |S_2| = 7, \quad \mathcal{M}_S = \{0, \dots, 7\} \times \{0, \dots, 6\}$



$$\mathcal{D} = \mathcal{M}_S, \quad \mathcal{C}(S, \mathcal{D}) = \{S \to \mathbb{F}\}$$

Examples of Downsets and Downset codes

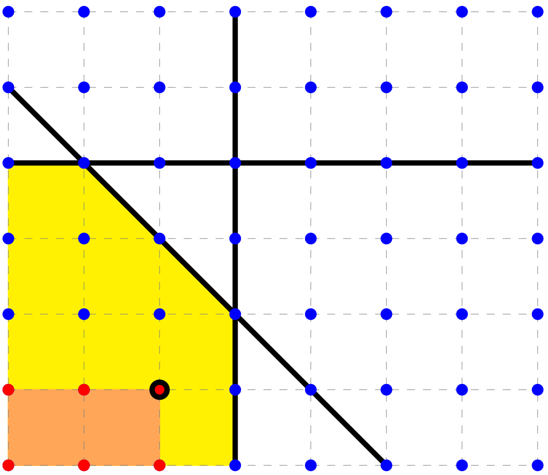
Eg. $n = 2$, $S = S_1 \times S_2$, $|S_1| = 8$, $|S_2| = 7$, $\mathcal{M}_S = \{0, \ldots, 7\} \times \{0, \ldots, 6\}$



$$\mathcal{D} = \{\mathbf{X}^\alpha \in \mathcal{M}_S : |\alpha| \leq 5\}, \quad \mathcal{C}(S, \mathcal{D}) = RM(S, 5)$$

Examples of Downsets and Downset codes

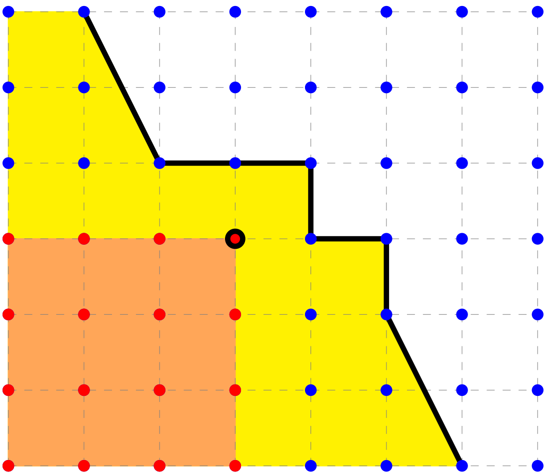
Eg. $n = 2, \quad S = S_1 \times S_2,\ |S_1| = 8,\ |S_2| = 7, \quad \mathcal{M}_S = \{0, \ldots, 7\} \times \{0, \ldots, 6\}$



$$\mathcal{D} = \{\mathbf{X}^\alpha \in \mathcal{M}_S : |\alpha| \leq 5,\ \alpha_1 \leq 3,\ \alpha_2 \leq 4\}, \quad \mathcal{C}(S, \mathcal{D})$$

Examples of Downsets and Downset codes

Eg. $n = 2$, $S = S_1 \times S_2$, $|S_1| = 8$, $|S_2| = 7$, $\mathcal{M}_S = \{0, \dots, 7\} \times \{0, \dots, 6\}$



$\mathcal{D}$, $\mathcal{C}(S, \mathcal{D})$

What is (unique) decoding?

Linear code $\mathcal{C}$

Definition (Hamming distance (metric))

$$\Delta(f, g) = |\{x \in S : f(x) \neq g(x)\}|, \quad f, g \in \mathcal{C}$$

The **Hamming weight** of $f \in \mathcal{C}$ is $\|f\| = |\mathrm{supp}(f)| = \Delta(f, 0)$.

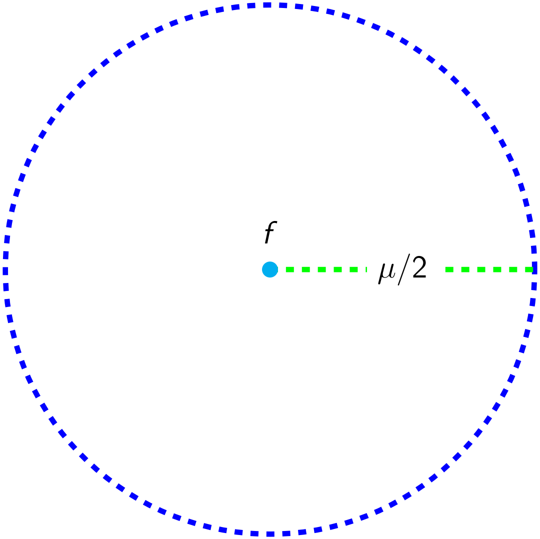
Definition (Minimum distance of a linear code)

$$\mu(\mathcal{C}) = \min\{\Delta(f, g) : f, g \in \mathcal{C},\ f \neq g\} = \min\{\|f\| : f \in \mathcal{C},\ f \neq 0\}$$

What is (unique) decoding?

$$\text{Linear code } \mathcal{C} \subseteq \mathbb{F}^k, \quad \mu = \mu(C), \quad f \in \mathbb{F}^k$$

$$B(f, \mu/2) = \{g \in \mathbb{F}^k : \Delta(f, g) < \mu/2\}$$

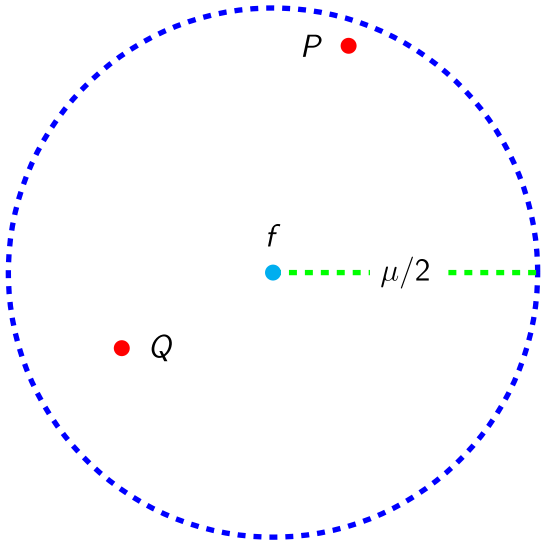


Case (I) $\quad B(f, \mu/2)$ with $B(f, \mu/2) \cap \mathcal{C} = \emptyset, \quad$ NO DECODING

What is (unique) decoding?

$$\text{Linear code } \mathcal{C} \subseteq \mathbb{F}^k, \quad \mu = \mu(C), \quad f \in \mathbb{F}^k, \quad P, Q \in \mathcal{C},\ P \neq Q$$

$$B(f, \mu/2) = \{g \in \mathbb{F}^k : \Delta(f, g) < \mu/2\}$$



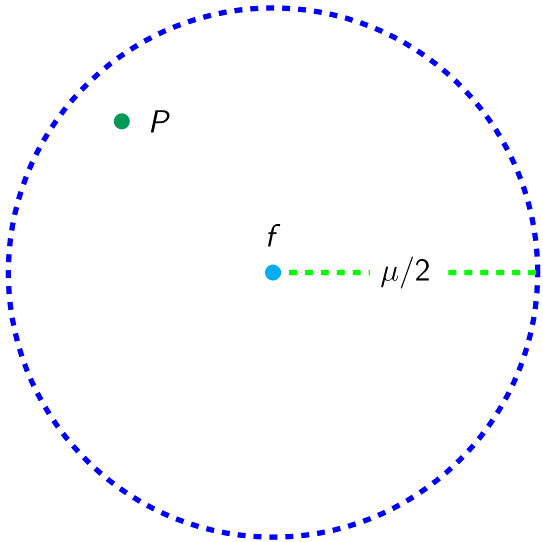
Case (II) $\quad B(f, \mu/2)$ with $B(f, \mu/2) \cap \mathcal{C} \supseteq \{P, Q\}$, $\quad$ NOT POSSIBLE

$$\Delta(P, Q) \leq \Delta(f, P) + \Delta(f, Q) < \mu/2 + \mu/2 = \mu \quad \text{NOT TRUE}$$

What is (unique) decoding?

Linear code $\mathcal{C} \subseteq \mathbb{F}^k, \quad \mu = \mu(C), \quad f \in \mathbb{F}^k, \quad P, Q \in \mathcal{C},\ P \neq Q$

$$B(f, \mu/2) = \{g \in \mathbb{F}^k : \Delta(f, g) < \mu/2\}$$



Case (III) $\quad B(f, \mu/2)$ with $B(f, \mu/2) \cap \mathcal{C} = \{P\}, \quad$ UNIQUE DECODING

Theorem (A template)

For 'appropriate' dimension n, field $\mathbb{F}$, finite grid $S \subseteq \mathbb{F}^n$ and the corresponding linear code $\mathcal{C} \subseteq \{S \to \mathbb{F}\}$, there is an algorithm which, given $f : S \to \mathbb{F}$ with the 'promise' that there exists a (unique) $P \in \mathcal{C}$ such that $\Delta(f, P) < \mu(\mathcal{C})/2$, returns P 'efficiently'.

	n	$\mathbb{F}$	S	$\mathcal{D}$	$\mathcal{C}(S, \mathcal{D})$
Reed (1954)	arbitrary arbitrary	$\mathbb{F}_2$ arbitrary	$\mathbb{F}_2^n$ $\{0,1\}^n$	$\mathcal{M}_{S,d}$ $\mathcal{M}_{S,d}$	$RM(\mathbb{F}_2^n, d)$ $RM(\{0,1\}^n, d)$
Forney (1966)[1]	1 1	$\mathbb{F}_q$ arbitrary	$\mathbb{F}_q$ arbitrary	$\mathcal{M}_{S,d}$ $\mathcal{M}_{S,d}$	$RS(\mathbb{F}_q, d)$ $RS(S, d)$
Berlekamp, Welch (1983)	1 1	$\mathbb{F}_q$ $\mathbb{F}_q$	$\mathbb{F}_q$ arbitrary	$\mathcal{M}_{S,d}$ $\mathcal{M}_{S,d}$	$RS(\mathbb{F}_q, d)$ $RS(\mathbb{F}_q, d)$
Kim, Kopparty (2017)	arbitrary	arbitrary	arbitrary	$\mathcal{M}_{S,d}$	$RM(S, d)$
Our result[2]	arbitrary	arbitrary	arbitrary	arbitrary	$\mathcal{C}(S, \mathcal{D})$

[1] with weights/uncertainties on words.

[2] joint work with Srikanth Srinivasan and Utkarsh Tripathi, both from Dept. of Mathematics, IIT Bombay.

Main Theorem

Theorem (Our Result)

There is a deterministic polynomial time algorithm such that, given a finite grid $S = S_1 \times \cdots \times S_n \subseteq \mathbb{F}^n$, a downset $\mathcal{D} \subseteq \mathcal{M}_S$, and $f : S \to \mathbb{F}$, the algorithm outputs $C \in \mathcal{C}(S, \mathcal{D})$ such that $\Delta(f, C) < \dfrac{\mu(S, \mathcal{D})}{2}$, if such a C exists. If such a C does not exist, then the algorithm outputs an arbitrary polynomial.

We will prove a slightly stronger version of the above involving a **weighted word** as input.

Weighted word and weighted distance

Definition

$$
\begin{aligned}
\text{Word} \quad & b : S \to \mathbb{F} \\
\text{Weighted word} \quad & (a, w) : S \to \mathbb{F} \times [0, 1] \\
\text{Weighted distance} \quad & \Delta((a, w), b) = \sum_{a(x)=b(x)} \left(\frac{w(x)}{2} \right) + \sum_{a(x) \neq b(x)} \left(1 - \frac{w(x)}{2} \right)
\end{aligned}
$$

Fact (Triangle Inequality)

Let $(a, w) : S \to \mathbb{F} \times [0, 1]$ be a weighted word and $b, c : S \to \mathbb{F}$ be words. Then

$$\Delta((a, w), b) + \Delta((a, w), c) \geq \Delta(b, c).$$

Further if $b, c \in \mathcal{C}(S, \mathcal{D})$ and $b \neq c$, then

$$\Delta((a, w), b) + \Delta((a, w), c) \geq \Delta(b, c) \geq \mu(S, \mathcal{D}).$$

In particular, $\Delta((a, w), b) < \mu(S, \mathcal{D})/2$ and $\Delta((a, w), c) < \mu(S, \mathcal{D})/2$ is not possible.

Main Theorem (Weighted version)

Theorem (Our Result)

There is a deterministic polynomial time algorithm such that, given a finite grid $S = S_1 \times \cdots \times S_n \subseteq \mathbb{F}^n$, a downset $\mathcal{D} \subseteq \mathcal{M}_S$, and $f : S \to \mathbb{F}$, the algorithm outputs $C \in \mathcal{C}(S, \mathcal{D})$ such that $\Delta(f, C) < \dfrac{\mu(S, \mathcal{D})}{2}$, if such a C exists. If such a C does not exist, then the algorithm outputs an arbitrary polynomial.

Theorem (Our Result, weighted version)

There is a deterministic polynomial time algorithm such that, given a finite grid $S = S_1 \times \cdots \times S_n \subseteq \mathbb{F}^n$, a downset $\mathcal{D} \subseteq \mathcal{M}_S$, and a **weighted word** $(a, w) : S \to \mathbb{F}$, *the algorithm outputs $C \in \mathcal{C}(S, \mathcal{D})$ such that $\Delta((a, w), C) < \dfrac{\mu(S, \mathcal{D})}{2}$, if such a C exists. If such a C does not exist, then the algorithm outputs an arbitrary polynomial.*

We will proceed by induction on n. The base case uses Forney's weighted Reed-Solomon decoder, mentioned earlier. For this talk, this decoder is a BLACK BOX.

Some more facts

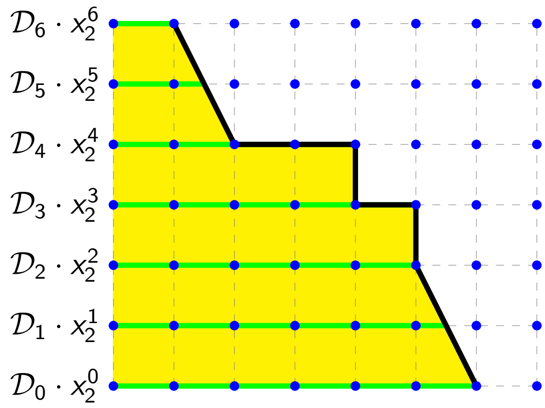
- Let $\mathcal{D} \subseteq \mathcal{M}_S$ be a downset and $\deg_n \mathcal{D} = \max\{\alpha_n : \alpha \in \mathcal{D}\}$. Define

$$\mathcal{D}_i = \{\beta \in \mathcal{M}_S : (\beta, i) \in \mathcal{D}\},$$

for $i \in \{0, \ldots, d = \deg_n \mathcal{D}\}$.
Then

(i) $\mathcal{D}_i$ is a downset, for all i, since $\mathcal{D}$ is a downset.

(ii) $\mathcal{D}_0 \supseteq \cdots \supseteq \mathcal{D}_d$.



- Let $\widetilde{S} = S_1 \times \cdots \times S_{n-1}$. For every $i \in \{0, \ldots, d = \deg_n \mathcal{D}\}$, we have

$$\mu(S, \mathcal{D}) \leq \mu(\widetilde{S}, \mathcal{D}_i) \cdot \mu(S_n, \{0, \ldots, i\}).$$

The Decoding Algorithm (An outline)

Base Case. $n = 1$. In this case, the algorithm is Forney's weighted decoding algorithm.
Induction Hypothesis. The weighted decoding algorithm works in the case $n = 2$.

Suppose $n = 3$.
Input: $(S_1 \times S_2 \times S_3, \mathcal{D}, (a, w))$, the grid, the downset and the weighted word.

Let $d = \deg_3 \mathcal{D}$.

Let the correct codeword be

$$C(x, y, z) = \sum_{j=0}^{d} Q_j(x, y) z^j, \quad \forall (x, y, z) \in S_1 \times S_2 \times S_3.$$

It is then enough to find $Q_j : S_1 \times S_2 \to \mathbb{F}$, $j = 0, \dots, d$. We will run $i \searrow d, \dots, 0$.

It is here, in finding Q_j-s that we use the inductive hypothesis, since the Q_j-s are 2-variate.
In order to do this, we need to find suitable weighted words.

The Decoding Algorithm (An outline)

Now consider a fixed i and suppose that at the i-th stage, the functions $Q_j(x, y)$, $j \searrow d, \ldots, i+1$ are known. When $i = d$, nothing is known (that's fine!). Let

$$a_i(x, y, z) = a(x, y, z) - \sum_{j=i+1}^{d} Q_j(x, y) z^j.$$

For every $(x, y) \in S_1 \times S_2$, define

$$a_{i,(x,y)}(z) = a_i(x, y, z), \quad w_{(x,y)}(z) = w(x, y, z).$$

Then we use Forney's algorithm for the one variable case. Apply the algorithm on the input $(S_3, \{0, \ldots, i\}, a_{i,(x,y)})$ to get the 'possibly correct 1-variate' word $G_{i,(x,y)} : S_3 \to \mathbb{F}$.

We have thus computed a 1-variate word $G_{i,(x,y)} : S_3 \to \mathbb{F}$ for each $(x, y) \in S_1 \times S_2$.

The Decoding Algorithm (An outline)

We now 'compare' weighted distances and determine the 'input' weighted words to be passed on to the (correct) 2-variable weighted decoder. Let

$$\Delta_{i,(x,y)} = \Delta(a_{i,(x,y)}, G_{i,(x,y)}), \quad \mu_i = \frac{\mu(S_n, \{0, \ldots, i\})}{2}.$$

If $\Delta_{i,(x,y)} < \mu_i$, let $\sigma_i(x,y) = [z^i](G_{i,(x,y)})$, $\delta_i(x,y) = \dfrac{\Delta_{i,(x,y)}}{\mu_i}$.
If $\Delta_{i,(x,y)} \geq \mu_i$, let $\sigma_i(x,y) = 0$, $\delta_i(x,y) = 1$.

We have thus computed a 2-variate weighted word $(\sigma_i, \delta_i) : S_1 \times S_2 \to \mathbb{F}$. We then give the input $(S_1 \times S_2, \mathcal{D}_i, (\sigma_i, \delta_i))$ to the 2-variate weighted decoder to get the output function $P_i : S_1 \times S_2 \to \mathbb{F}$.

Since by inductive assumption, the algorithm is correct for the 2-variate case, we have $P_i = Q_i$. (It is a routine case analysis to check that $\Delta(P_i, Q_i) < \mu(S_1 \times S_2, \mathcal{D}_i)/2$. This is why the correct decoder gives the correct output!)

Running through $i \searrow d, \ldots, 0$ gives the correct codeword $C(x,y,z) = \sum_{j=0}^{d} Q_j(x,y)z^j$.

Thank You!