$$l^*(s) = \int_0^1 \int_{1/2}^1 \left[\begin{array}{c} (1+\tau_i)\frac{1}{a_{z,i}}\left(\frac{\hat{p}_{z,i}}{\hat{P}_i(s)}\right)^{-\gamma}\left(\frac{\hat{P}_i(s)}{\hat{P}(s)}\right)^{-\theta} C(s) + \frac{1}{a_{z,i}}\left(\frac{\hat{p}^*}{\hat{P}_i^*(s)}\right)^{-\gamma}\left(\frac{\hat{P}_i^*(s)}{\hat{P}^*(s)}\right)^{-\theta} C^*(s) \\ +\left(\kappa_i \mathcal{I}\left(\hat{p}_{z,i} \neq \hat{p}_{-1,z,i}\right) + \kappa_i \mathcal{I}\left(\hat{p}^*_{z,i} \neq \hat{p}^*_{-1,z,i}\right)\right)(1-\lambda_i) \end{array} \right] didz +$$

4. Computation

We next discuss the solution method used to solve the system of function equations above.

A. *Firm's problem* Given $C(s), R(s), l(s), \hat{W}(s), \hat{e}(s), \hat{P}(s), \hat{P}_i(s_i, s)$, the firm's problem in each of the 2 markets is characterized by a system of 2 functional equations in V^a and V^n. Solving this system of functional equations is complicated by a) the large dimensionality of the state-space (7 state variables in the simplest version of the problem), as well as b) the high curvature of the value functions, inherited from the curvature of profit functions for high values of γ (the elasticity of substitution across varieties of a good). The non-convexity that arises from the fixed adjustment cost requires very good approximants for us to be able to accurately characterize firm decision rules.

To deal with the curse of dimensionality, we employ Smolyak sparse grid interpolation, instead of the usual tensor product methods of extending univariate interpolation methods to multivariate problems. To deal with the curvature of the value functions, we redefine our state variables and work with functions of the original state variables that reduce the curvature of the state space.

B. *Smolyak interpolation* We provide a very brief description of the method. See Kubler and Kruger (2004) or Winschel (2005) for a more detailed discussion.

Univariate functional approximation techniques involve approximating an analyticaly intractable function $f : [-1,1] \to R$, that satisfies $g(f(x)) = 0$, with an approximant $\hat{f}(x)$ that is a linear combination of low-order polynomials:

$$\hat{f}(x) = \sum_{i=1}^{n} c_i \phi_i(x)$$

where ϕ_i are the basis functions (we employ Chebyshev polynomials), and the coefficients c_i are found by requiring that the approximant satisfy $g(\hat{f}(x_k)) = 0$ at $k \geq n.$ nodes in the state-space.

The typical method of extending univariate methods to multivariate problems where the state-space is $d-$dimensional is to employ tensor products of univariate basis functions ϕ_i^j, $j = 1...d$, $i = 1...n_j$ and to solve for the unknown coefficients on a Cartesian grid of the univariate nodes:

$$\hat{f}(x_1, ..., x_d) = \sum_{i_1=1}^{n_1} ... \sum_{i_d=1}^{n_d} c_{i_1...i_d} \phi_{i_1}^1(x_1)...\phi_{i_d}^d(x_d)$$

For example, to approximate an unknown function $f(x,y)$ using univariate basis functions $\{1,\ x, x^2\}$ and $\{1, y, y^2\}$, at nodes $X = \{X^1, X^2, X^3\}$ and $Y = \{Y^1, Y^2, Y^3\}$, one forms the multivariate basis functions, by taking the tensor product of the univariate functions: $\{1,\ x, x^2\} \otimes \{1,\ y, y^2\} = \{1, x, x^2, y, xy, x^2y, y^2, y^2x, y^2x^2\}$ and solves for the 9 unknown coefficients by requiring that $g(\hat{f}(x,y)) = 0$ be satisfied at $X \times Y$.

The drawback of this method is that it badly suffers from the curse of dimensionality. In our particular problem, even if we were to allow only 3 nodes along each of the 7 dimensions (i.e., at most a quadratic in each dimension), we would have to solve a non-linear system of equations in 3^7=2187 unknown coefficients.

Sparse grid methods deal with the curse of dimensionality by truncating some of the high-order polynomials (in the example above one could eliminate polynomials of order greater than 2: x^2y, y^2x, y^2x^2) and solving the functional equation at a sparse grid of points by throwing out some of the nodes used in tensor-product methods.

The Smolyak algorithm we employ is one particular method of eliminating high-order polynomials. Originally suggested by Smolyak (1963), the method has recently enjoyed popularity due to an optimality proof by Barthelmann, Novak and Ritter (2000).

The discussion below is terse, but sufficiently complete to allow the reader to reproduce our results. Detailed discussions can be found in the references cited above. We illustrate how the method works with some simple examples.

Let $m_i = 2^i - 1$ for $i > 1$, 1 for $i = 1$. Denote the uni-dimensional nodes in dimension $j = 1...d$ of level i by $X_i^j = \{X_1^j, ..., X_{m_i}^j\}$.Instead of using a Cartesian product of univariate nodes, we construct nodes for Smolyak interpolation using the following formula, where d is the dimensionality of the state-space, $q > d$ is an integer that determines the accuracy of the approximation, and $H_{q,d}$ is the set of nodes used for interpolation.

$$H_{q,d} = \bigcup_{q-d+1 \leq |\mathbf{i}| \leq q} \left(X_{i_1}^1 \times ... \times X_{i_d}^d \right)$$

where $|\mathbf{i}| = i_1 + ... + i_d$.Similarly, letting ϕ_i^j denote the basis function in the $j - th$ dimension of level i, the approximant is constructed using:

$$\hat{f}(x_1, ..., x_d) = (-1)^{q-|\mathbf{i}|} \left(\begin{array}{c} d-1 \\ \\ q-|\mathbf{i}| \end{array} \right)_{q-d+1 \leq |\mathbf{i}| \leq q} \left(\phi^1_{i_1} \otimes ... \otimes \phi^d_{i_d}\right)$$

Intuitively, we throw out all products of univariate basis functions that would give rise to high-order polynomials: (this is the requirement that $|\mathbf{i}| \leq q$). In our earlier example with $d = 2$,

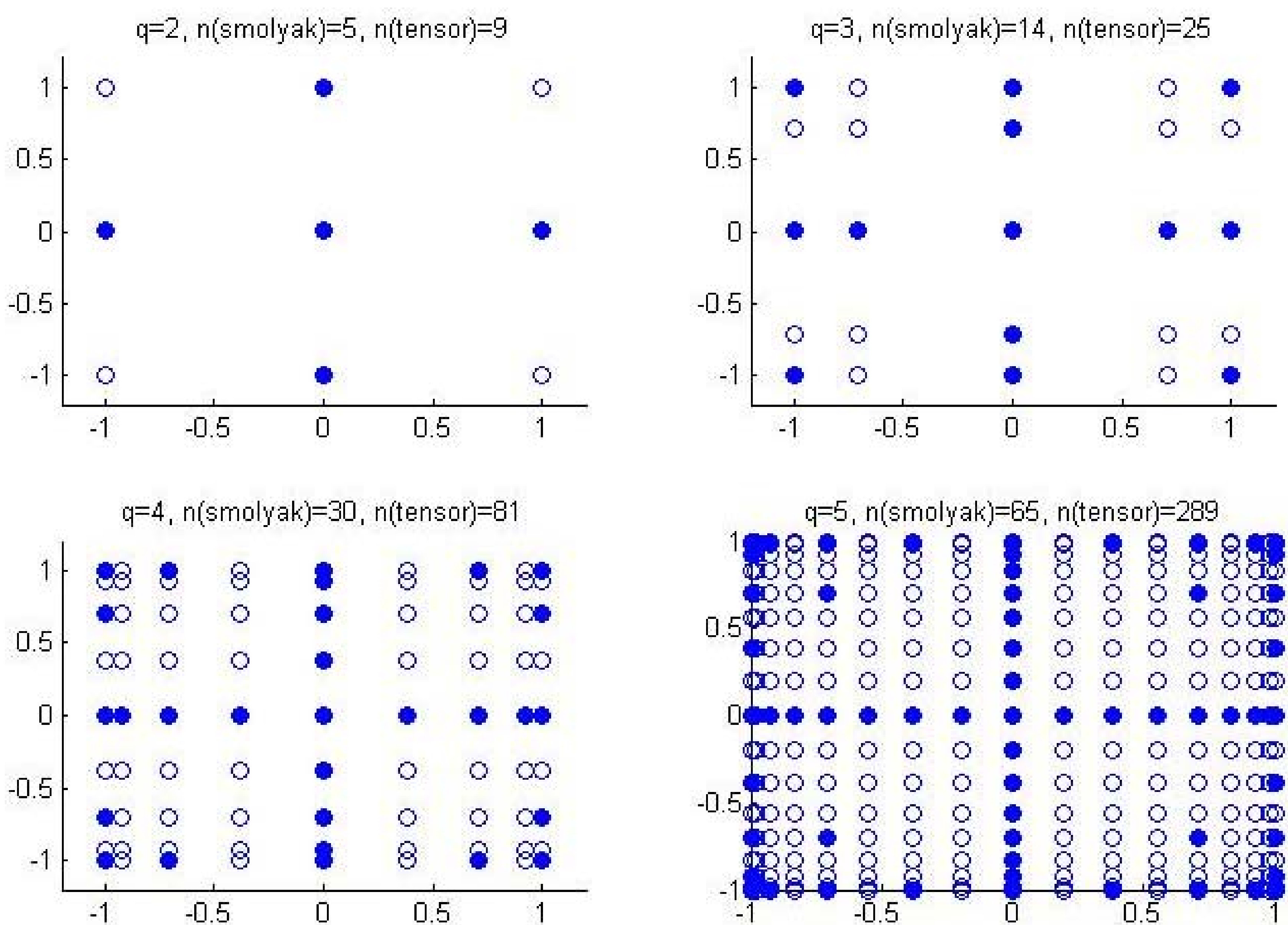
$\phi^x_1 = \{1\}, \phi^x_2 = \{1, x, x^2\}, \phi^x_3 = \{1, x, x^2, x^3, x^4\}$ and $\phi^y_1 = \{1\}, \phi^y_2 = \{1, y, y^2\}, \phi^y_3 = \{1, y, y^2, y^3, y^4\}$

The basis functions that survive the filter above are thus:

$q = 3: \{\underbrace{1}_{i_1+i_2=2}, \underbrace{x, x^2, y, y^2}_{i_1+i_2=3}\}$

$q = 4: \{\underbrace{1}_{i_1+i_2=2}, \underbrace{x, x^2, y, y^2}_{i_1+i_2=3}, \underbrace{xy, x^2y, xy^2, x^2y^2, x^3, x^4, y^3, y^4}_{i_1+i_2=4}\}$

As for the nodes, the figures below illustrates the choice of nodes for several values of q.


q=2, n(smolyak)=5, n(tensor)=9
q=3, n(smolyak)=14, n(tensor)=25
q=4, n(smolyak)=30, n(tensor)=81
q=5, n(smolyak)=65, n(tensor)=289
-1
-0.5
0
0.5
1