Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Mitigation of Curse of Dimensionality in Dynamic Programming ⋆

D. Li * **Q. Wang** ** **J. Wang** *** **Y. R. Yao** ****

* *Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: dli@se.cuhk.edu.hk)*
** *Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong*
*** *Department of Management Science and Engineering, Qingdao University, Qingdao, Shandong 266071, China*
**** *Department of Mathematics, Shanghai University, Baoshan, Shanghai 200444, China*

Abstract: Dynamic programming, one of the most powerful solution methodologies to achieve optimality for separable optimization problems, suffers heavily from the notorious "curse of dimensionality", which prevents its direct applications when the dimension of the state space is high. By aggregating multiple constraints into a single surrogate constraint, the surrogate constraint formulation offers an ideal platform for powerful utilization of dynamic programming, although often with a price of a presence of duality gap. In this paper, we propose a novel convergent dynamic programming algorithm by integrating a domain cut scheme with the surrogate constraint formulation, thus enabling elimination of the duality gap gradually in the solution process.

1. INTRODUCTION

By invoking a decomposition scheme based on the ground-breaking principle of optimality, dynamic programming pioneered by Richard Bellman in 1950's is one of the most powerful solution methodologies for separable optimization problems. The past four decades have witnessed great success of applications of dynamic programming and also many extensions of it, for example, nonserial dynamic programming (Esogbue (1972)), multi-objective dynamic programming (Li and Haimes (1987), Liao and Li (2002)), and nonseparable dynamic programming (Li and Haimes (1990)).

Although dynamic programming is a universal solution scheme for separable optimization problems, it suffers heavily from the notorious "curse of dimensionality" as named by Bellman himself, which prevents a direct application of dynamic programming when the state space is large.

Mitigating the curse of dimensionality in dynamic programming has been a challenging research task in front of the control and optimization community for many years. There exist a few papers concerning mitigation of curse of dimensionality for discrete dynamic programming. Recognizing a relationship between the optimal solutions and the efficient solutions in the constraint space, a hybrid method was developed in Dyer et al. (1995), Korner (1989), Marsten and Morin (1978) and Morin and Esogbue (1974) with a purpose to fathom in the solution process inefficient and incomplete feasible solutions by bounds and dominance rules. Many attempts have been made to mitigate the curse of dimensionality of dynamic programming in its control applications. A successive approximation technique was proposed in Larson (1965) and Larson and Korsak (1970) in which a single state is perturbed each time on the incumbent trajectory, resulting in a sequence of scalar-state dynamic programming problems. Differential dynamic programming developed in Jacobson and Mayne (1970), Liao and Shoemaker (1991), Mayne (1966) and Yakowitz and Rutherford (1984) is a second-order method that successively improves the incumbent trajectory under a convexity assumption. The idea of region reduction was adopted in Luus (1998) by successively refining a coarse grid assignment of the state space. Note that the curse of dimensionality disappears when an analytical form of the cost-to-go function can be achieved. Thus, different numerical methods, such as linear and spline interpolation in Johnson et al. (1993) and neural computing in Bertsekas and Tsitsiklis (1996), have been suggested in the literature to approximate the cost-to-go by an analytical form. The state-of-the-art of the battle against the curse of dimensionality is still far below a satisfactory level, in terms of the present computational power for high-dimensional dynamic programming.

We consider dynamic programming in this paper as a solution method for the following general class of multiply constrained separable integer programming problems:

$$(P) \qquad \min\ f(x) = \sum_{j=1}^{n} f_j(x_j)$$

⋆ This work was supported in part by the Research Grants Council of Hong Kong, grant no. 414207.

$$\text{s.t.} \quad g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j) \le b_i, \; i = 1, \ldots, m,$$
$$x \in X = X_1 \times X_2 \times \cdots \times X_n,$$

where f_j's and g_{ij}'s are real-valued functions defined on $\mathbb{R}$, and all X_j's are finite integer sets in $\mathbb{R}$. Problem (P) has a wide variety of applications, including resource allocation problems and nonlinear multi-dimensional knapsack problems (see Li and Sun (2006) and the references therein). At the same time, problem (P) covers very general situations of nonlinear integer programming problems as no additional property such as convexity, concavity, monotonicity or differentiability is assumed in (P).

To apply dynamic programming to (P), we first introduce a stage variable k, $0 \le k \le n$, and a state vector at stage k, $s_k \in \mathbb{R}^m$, satisfying the following recursive equation:

$$s_{k+1} = s_k + g^k(x_k), \; k = 1, \ldots, n-1,$$

with an initial condition $s_1 = 0$, where $g^k(x_k) = (g_{1k}(x_k), \ldots, g_{mk}(x_k))^T$. Assume that the constraints are integer-valued, then we only need to consider integer points in the state space. Furthermore, the feasible region of the state vector at stage k with $2 \le k \le n+1$ can be confined to $\underline{s}_k \le s_k \le \bar{s}_k$, where for $i = 1,\ldots,m$,

$$(\underline{s}_k)_i = \sum_{t=1}^{k-1} \min_{x_t \in X_t} g_{it}(x_t),$$
$$(\bar{s}_k)_i = \min\{\sum_{t=1}^{k-1} \max_{x_t \in X_t} g_{it}(x_t), b_i - \sum_{t=k}^{n} \min_{x_t \in X_t} g_{it}(x_t)\}.$$

For a given state s at stage k, $1 \le k \le n$, we define the cost-to-go function as follows,

$$t_k(s) = \min \sum_{j=k}^{n} f_j(x_j),$$
$$\text{s.t.} \quad s + \sum_{j=k}^{n} g^j(x_j) \le b,$$
$$x_j \in X_j, \; j = k, \ldots, n.$$

It is obvious that $v(P) = t_1(0)$, where $v(\cdot)$ is the optimal value of an optimization problem $(\cdot)$. Based on Bellman's principle of optimality, the cost-to-go function satisfies the following backward recursive relation for $k = n-1, \ldots, 1$,

$$t_k(s) = \min_{x_k \in X_k} \{f_k(x_k) + t_{k+1}(s + g^k(x_k))\}$$

with boundary condition

$$t_n(s) = \min_{x_n \in X_n} \{f_n(x_n) \mid s + g^n(x_n) \le b\}.$$

The backward dynamic programming starts at $k = n - 1$ and moves backwards to $k = n-2, \ldots, 1$. The cost-to-go functions are calculated recursively for every s at each stage k between $\underline{s}_k$ and $\bar{s}_k$ and finally stops at $s_1 = 0$. The tracing process is then carried out in a forward way to identify the optimal solution(s) of (P).

The above classical dynamic programming algorithm suffers from the "curse of dimensionality" even when the number of constraints is relatively large. More specifically, at each stage k, we need to calculate the cost-to-go at $\prod_{i=1}^{m}[(\bar{s}_k)_i - (\underline{s}_k)_i + 1]$ possible states, which grows exponentially with respect to m.

In this paper, we develop a novel solution framework to overcome the curse of dimensionality in dynamic programming. More specifically, as dynamic programming remains an efficient solution scheme for singly constrained problems, we build up our convergent dynamic programming algorithms on the platform of a surrogate constraint formulation. Domain cut scheme is then introduced to gradually remove "active" infeasible solutions that attain optimality in the surrogate constraint formulation, thus reducing the duality gap successively and eventually eliminating it.

2. SURROGATE CONSTRAINT FORMULATION

The surrogate constraint formulation (Glover (1965)) is formed by aggregating multiple constraints into a single surrogate constraint,

$$(P_\mu) \qquad \min \; f(x)$$
$$\text{s.t.} \quad \mu^T(g(x) - b) \le 0, \; x \in X,$$

where $\mu = (\mu_1, \ldots, \mu_m)^T \in \mathbb{R}^m_+$ is a vector of surrogate multipliers, $g(x) = (g_1(x), \ldots, g_m(x))^T$ and $b = (b_1, \ldots, b_m)^T$. The surrogate constraint formulation provides a platform for battling against curse of dimensionality.

Denote by S and $S(\mu)$ the feasible regions of (P) and (P_μ), respectively. Since, for any $\mu \in \mathbb{R}^m_+$, $S(\mu)$ is an enlargement of the feasible set of the primal problem, i.e., $S \subseteq S(\mu)$, (P_μ) is a relaxation of the primal problem (P). By weak duality, solving (P_μ) offers a lower bound of the optimal value of (P):

$$v(P_\mu) \le v(P), \quad \forall \; \mu \; \in \; \mathbb{R}^m_+.$$

The surrogate dual is to seek the best lower bound generated by (P_μ):

$$(D_S) \qquad \max_{\mu \in \mathbb{R}^m_+} \; v(P_\mu).$$

In general, the quality of a dual scheme should be judged by two measures. The first is how easier the relaxation problem can be solved when compared with the primal problem. The second is how tight the best lower bound (optimal dual value) can be, in other words, how small the duality gap can be. Note that surrogate constraint formulation (P_μ) is a singly constrained separable integer programming problem which can be efficiently solved by dynamic programming. Moreover, it can be shown that the lower bound generated by the surrogate dual (D_S) is tighter than the bound by the conventional Lagrangian dual (see Li and Sun (2006)).

It is easy to see that if an optimal solution to (P_μ) for some $\mu \in \mathbb{R}^m_+$ happens to be feasible to (P), then it is also optimal to the primal problem (P), i.e., strong duality holds. However, the presence of attaining strong duality is rare in surrogate dual search. Achieving the primal feasibility for an optimal solution to some revised surrogate relaxation problem via domain cut is the primal goal of the research presented in this paper.

3. DOMAIN CUT UNDER SURROGATE CONSTRAINT FORMULATION

The motivation behind the proposed solution scheme is to gradually remove some "active" infeasible solutions of (P) that attain optimal positions in the surrogate constraint formulation. More specifically, a domain cut approach is adopted to cut off sub-domains in the feasible region of the surrogate formulation (P_μ) that contain "active" infeasible solutions and "non-promising" feasible solutions from further consideration.

Let α and β be integer vectors in $\mathbb{R}^n$. Denote $\langle\alpha_j,\beta_j\rangle = \{x_j \mid \alpha_j \le x_j \le \beta_j,\ x_j \text{ integer}\}$ and $\langle\alpha,\beta\rangle = \Pi_{j=1}^n\langle\alpha_j,\beta_j\rangle$. Let $\lfloor t\rfloor$ denote the maximum integer less than or equal to t and $\lceil t\rceil$ the minimum integer greater than or equal to t.

When the duality gap is nonzero, domain cut can be implemented to cut off some infeasible integer sub-boxes whose objective values dominate the feasible region of the surrogate constraint formulation. Repeatedly carrying out such a procedure gradually reduces the duality gap and eventually eliminates it. The solution of this iterative process converges to the optimal solution of the primal problem. The following theorem gives conditions under which the above domain cut procedure can be applied.

Theorem 1. Let $\tilde{x}$ be a solution to (P_μ) on $\langle\alpha,\beta\rangle$. If $\tilde{x}$ is infeasible to (P), more specifically, $g_k(\tilde{x}) > b_k$ for some $k \in \{1,\ldots,m\}$, then the following hold.

(i) If f is concave, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} < 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} > 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} = 0$, contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

(ii) If f is a convex quadratic function taking the following form: $f(x) = \sum_{j=1}^n(\frac{1}{2}c_jx_j^2 + d_jx_j)$ with all $c_j > 0$, $j = 1, \ldots, n$, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \lceil -\frac{d_i}{c_i} - |\tilde{x}_i + \frac{d_i}{c_i}|\rceil$ and $\delta_i = \lceil -\frac{d_i}{c_i} + |\tilde{x}_i + \frac{d_i}{c_i}|\rceil$ contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

(iii) If f is a concave quadratic function taking the following form: $f(x) = \sum_{j=1}^n(\frac{1}{2}c_jx_j^2 + d_jx_j)$ with all $c_j < 0$, $j = 1, \ldots, n$, then the optimal solution can only be in the following integer region $\langle\rho,\sigma\rangle \setminus \langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\rho_i = \lceil -d_i/c_i - \sqrt{\left|2(f(\tilde{x}) + \sum_{j=1}^n d_j^2/(2c_j))/c_i\right|}\rceil$, $\sigma_i = \lceil -d_i/c_i + \sqrt{\left|2(f(\tilde{x}) + \sum_{j=1}^n d_j^2/(2c_j))/c_i\right|}\rceil$, and $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} < 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} > 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} = 0$.

(iv) If f is a monotone function of x, i.e., for all i, $i = 1, \ldots, n$, f is either increasing or decreasing with respect to x_i, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} < 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} > 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} = 0$, contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

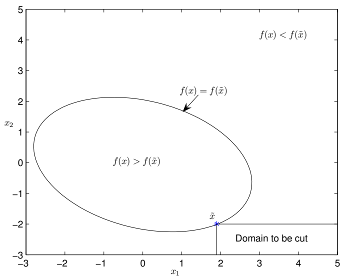

Fig. 1. Domain cut when f is concave

(v) If g_k is convex, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} > 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} < 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} = 0$, contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

(vi) If g_k is a convex quadratic function of x taking the following form: $g_k(x) = \sum_{j=1}^n(\frac{1}{2}c_{kj}x_j^2 + d_{kj}x_j)$ with all $c_{kj} > 0$, $j = 1, \ldots, n$, then the optimal solution $\langle\alpha,\beta\rangle$ can only be in the following integer region $\langle\rho,\sigma\rangle \setminus \langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\rho_i = \lceil -d_{ki}/c_{ki} - \sqrt{\left|2(g_k(\tilde{x}) + \sum_{j=1}^n d_{kj}^2/(2c_{kj}))/c_{ki}\right|}\rceil$, $\sigma_i = \lceil -d_{ki}/c_{ki} + \sqrt{\left|2(g_k(\tilde{x}) + \sum_{j=1}^n d_{kj}^2/(2c_{kj}))/c_{ki}\right|}\rceil$ and $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} > 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} < 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} = 0$.

(vii) If g_k is a concave quadratic function of x taking the following form: $g_k(x) = \sum_{j=1}^n(\frac{1}{2}c_{kj}x_j^2 + d_{kj}x_j)$ with all $c_{kj} < 0$, $j = 1, \ldots, n$, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \lceil -\frac{d_{ki}}{c_{ki}} - |\tilde{x}_i + \frac{d_{ki}}{c_{ki}}|\rceil$ and $\delta_i = \lceil -\frac{d_{ki}}{c_{ki}} + |\tilde{x}_i + \frac{d_{ki}}{c_{ki}}|\rceil$, contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

(viii) If g_k is a monotone function of x, then the following integer subbox $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} > 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} < 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} = 0$, contains no feasible solution of (P) and can be removed from $\langle\alpha,\beta\rangle$.

Proof. We will give separate proofs for all of the above eight cases.

(i) When f is concave, the set $\{x \in \langle\alpha,\beta\rangle \mid f(x) \ge f(\tilde{x})\}$ is a convex set as shown in Figure 1, outside of which all points have an objective value strictly less than $f(\tilde{x})$. Note that $f(\tilde{x})$ is a lower bound of $v(P)$ on $\langle\alpha,\beta\rangle$ and, by the weaker duality, no point outside of $\{x \in \langle\alpha,\beta\rangle \mid f(x) \ge f(\tilde{x})\}$ can be optimal. Based on the sign of the normal vector of f at $\tilde{x}$, the box $\langle\gamma,\delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} < 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} > 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} = 0$, is outside of $\{x \in \langle\alpha,\beta\rangle \mid f(x) \ge f(\tilde{x})\}$ and can be removed.

(ii) Consider the following ellipse contour of f:

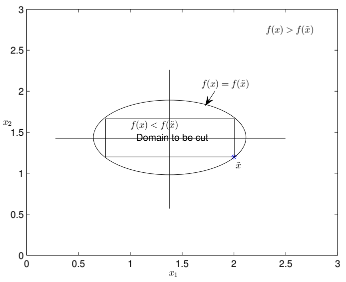

Fig. 2. Domain cut when f is convex and quadratic

$$f(x) = \sum_{j=1}^{n}[(1/2)c_j x_j^2 + d_j x_j] = f(\tilde{x}). \tag{1}$$

Clearly, the center of ellipse (1) is

$$o = (-d_1/c_1, \ldots, -d_n/c_n)^T. \tag{2}$$

See Figure 2. Let $E(\tilde{x})$ be the ellipsoid formed by the above ellipse contour. Since f is convex, all points inside $E(\tilde{x})$ possess an objective value smaller than $f(\tilde{x})$. By the weaker duality, no point inside $E(\tilde{x})$ can be feasible. By the symmetry of $E(\tilde{x})$, the integer box $\langle\gamma, \delta\rangle$, with

$$\gamma = (\lceil o_1 - |\tilde{x}_1 - o_1|\rceil, \ldots, \lceil o_n - |\tilde{x}_n - o_n|\rceil)^T, \tag{3}$$
$$\delta = (\lfloor o_1 + |\tilde{x}_1 - o_1|\rfloor, \ldots, \lfloor o_n + |\tilde{x}_n - o_n|\rfloor)^T, \tag{4}$$

is inside $E(\tilde{x})$ and can be removed.

(iii) Consider the ellipse contour of f given in (1) with the center specified in (2). Note that the length of the i-th axis of the ellipse contour is

$$2r_i = 2\sqrt{\left|2(f(\tilde{x}) + \sum_{j=1}^{n} d_j^2/(2c_j))/c_i\right|}. \tag{5}$$

See Figure 3. Let $E(\tilde{x})$ be the ellipsoid formed by the above ellipse contour. Since f is concave, all points outside $E(\tilde{x})$ possess an objective value smaller than $f(\tilde{x})$. The minimum rectangle that encloses the ellipsoid $E(\tilde{x})$ is $[\rho, \sigma]$ with

$$\rho = (o_1 - r_1, \ldots, o_n - r_n)^T,$$
$$\sigma = (o_1 + r_1, \ldots, o_n + r_n)^T,$$

and the optimal solution cannot be outside of this minimum rectangle. We can further cut off integer subbox $\langle\gamma, \delta\rangle$ from $\langle\rho, \sigma\rangle$ based on the argument given in Item (i).

(iv) If f is monotone, then any point in the subbox $\langle\gamma, \delta\rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} < 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} > 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial f(\tilde{x})}{\partial x_i} = 0$, has an objective level not greater than $f(\tilde{x})$. Since $f(\tilde{x})$ is a lower bound of problem (P) on $\langle\alpha, \beta\rangle$, no point inside $\langle\gamma, \delta\rangle$ can be optimal. See Figure 4.

(v) When g_k is convex, the set $\{x \in \langle\alpha, \beta\rangle \mid g_k(x) \le g_k(\tilde{x})\}$ is a convex set as shown in Figure 5, outside of which all points have a g_k value strictly larger than $g_k(\tilde{x})$. In other words, no point outside of $\{x \in \langle\alpha, \beta\rangle \mid g_k(x) \le g_k(\tilde{x})\}$ can be feasible. Based on the sign of the normal vector

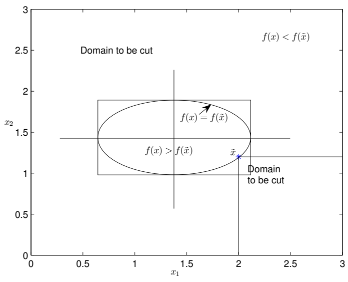

Fig. 3. Domain cut when f is concave and quadratic

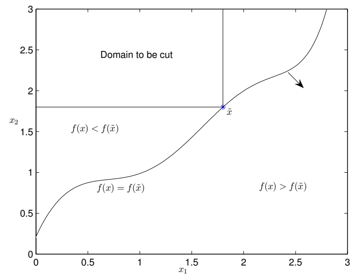

Fig. 4. Domain cut when f is monotone

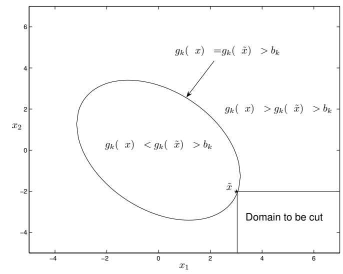

Fig. 5. Domain cut when g_k is convex

of g_k at $\tilde{x}$, the box $\langle\gamma, \delta\rangle$, with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} > 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} < 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} = 0$, is outside of $\{x \in \langle\alpha, \beta\rangle \mid g_k(x) \le g_k(\tilde{x})\}$ and can be removed.

(vi) Consider the following ellipse contour of g_k:

$$\sum_{j=1}^{n}[(1/2)c_{kj} x_j^2 + d_{kj} x_j] = g_k(\tilde{x}). \tag{6}$$

Clearly, the center of ellipse (6) is

$$o = (-d_{k1}/c_{k1}, \ldots, -d_{kn}/c_{kn})^T \tag{7}$$

and the length of the j-th axis of ellipse (6) is

$$2r_j = 2\sqrt{\left|2(g_k(\tilde{x}) + \sum_{l=1}^{n} d_{kl}^2/(2c_{kl}))/c_{kj}\right|}. \tag{8}$$

See Figure 6. Let $E(\tilde{x})$ be the ellipsoid formed by the above ellipse contour. Since g_k is convex, all points outside $E(\tilde{x})$ possess a g_k value larger than $g_k(\tilde{x})$. The minimum rectangle that encloses the ellipsoid $E(\tilde{x})$ is $[\rho, \sigma]$ with

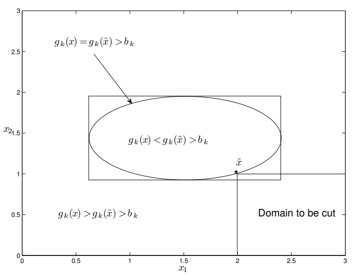

Fig. 6. Domain cut when g_k is convex and quadratic

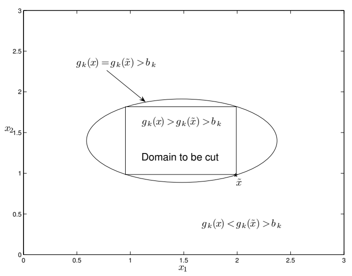

Fig. 7. Domain cut when g_k concave and quadratic

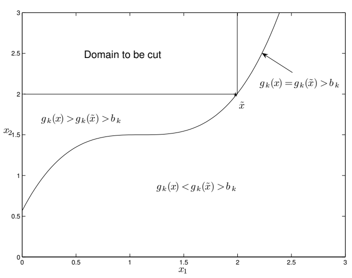

Fig. 8. Domain cut when g_k monotone

$$\rho = (o_1 - r_1, \ldots, o_n - r_n)^T,$$
$$\sigma = (o_1 + r_1, \ldots, o_n + r_n)^T,$$

and the optimal solution cannot be outside of this minimum rectangle. We can further cut off integer subbox $\langle \gamma, \delta \rangle$ from $\langle \rho, \sigma \rangle$ based on the argument given in Item (v).

(vii) Consider the ellipse contour of g_k given in (6) with the center specified in (7). See Figure 7. Let $E(\tilde{x})$ be the ellipsoid formed by the above ellipse contour. Since g_k is concave, all points inside $E(\tilde{x})$ possess a g_k value larger than $g_k(\tilde{x})$, thus all infeasible. By the symmetry of $E(\tilde{x})$, the integer box $\langle \gamma, \delta \rangle$, with

$$\gamma = (\lceil o_{k1} - |\tilde{x}_1 - o_{k1}| \rceil, \ldots, \lceil o_{kn} - |\tilde{x}_n - o_{kn}| \rceil)^T, \quad (9)$$
$$\delta = (\lfloor o_{k1} + |\tilde{x}_1 - o_{k1}| \rfloor, \ldots, \lfloor o_{kn} + |\tilde{x}_n - o_{kn}| \rfloor)^T, \quad (10)$$

is inside $E(\tilde{x})$ and can be removed.

(viii) If g_k is monotone, then any point in the subbox $\langle \gamma, \delta \rangle$ with, for $i = 1, \ldots, n$, $\gamma_i = \tilde{x}_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} > 0$, $\gamma_i = \alpha_i$ and $\delta_i = \tilde{x}_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} < 0$, and $\gamma_i = \alpha_i$ and $\delta_i = \beta_i$ if $\frac{\partial g_k(\tilde{x})}{\partial x_i} = 0$, has a g_k level not less than $g_k(\tilde{x})$, thus infeasible. See Figure 8. $\square$

Note that cutting out $\langle \gamma, \delta \rangle$ from $\langle \alpha, \beta \rangle$ results in a non-box domain which can be expressed as a union of multiple integer boxes. A key issue is how to partition a non-rectangular domain into a union of integer boxes such that the surrogate constraint dynamic programming can be applied to every newly generated integer sub-box after a cutting process. We have the following result from Li and Sun (2006).

Lemma 2. Let $A = \langle \alpha, \beta \rangle$ and $B = \langle \gamma, \delta \rangle$, where α, β, γ, $\delta \in \mathbb{Z}^n$ and $\alpha \le \gamma \le \delta \le \beta$. Then $A \setminus B$ can be partitioned into at most $2n$ integer boxes.

$$\begin{aligned} & A \setminus B \\ & = \left\{ \cup_{j=1}^n \left(\Pi_{i=1}^{j-1} \langle \alpha_i, \delta_i \rangle \times \langle \delta_j + 1, \beta_j \rangle \times \Pi_{i=j+1}^n \langle \alpha_i, \beta_i \rangle \right) \right\} \\ & \cup \left\{ \cup_{j=1}^n \left(\Pi_{i=1}^{j-1} \langle \gamma_i, \delta_i \rangle \times \langle \alpha_j, \gamma_j - 1 \rangle \times \Pi_{i=j+1}^n \langle \alpha_i, \delta_i \rangle \right) \right\}. \end{aligned}$$

Since problem (P) is separable, we can apply Theorem 1 separately when f or g_k satisfies one condition with respect to x_i and satisfies another condition with respect to x_j, with $j \neq i$.

4. CONVERGENT SURROGATE CONSTRAINT DYNAMIC PROGRAMMING

Integrating the domain cut scheme with the surrogate constraint formulation under a framework of branch-and-bound, we now formally describe the solution algorithm.

Step 0 (Initialization). Select a surrogate multiplier μ and use dynamic programming to solve (P_μ). Let x^0 be the solution to (P_μ). If x^0 is feasible, then x^0 is the optimal solution of (P) and stop. Otherwise, calculate $f(x^0)$. Let $X^0 = X$, $k = 0$, $x_{opt} = \emptyset$ and $f_{opt} = -\infty$.

Step 1 (Sub-Domain Selection) Select an integer subbox X^{kj} from X^k with the smallest objective value of (P_μ), $f(x^{kj})$. Let $X^k = X^k \setminus X^{kj}$.

Step 2 (Cut and Partition) Cut out from X^{kj} certain integer boxes of infeasible solutions that include x^{kj} using one of the formula in Theorem 1 and partition the remaining domain, Z^k, into a union of integer sub-boxes.

Step 3 (Evaluation) Solve (P_μ) on every integer subbox in Z^k by using dynamic programming. Remove all the integer subboxes from Z^k whose solution is feasible in (P). Update x_{opt} and f_{opt} if a feasible solution found possesses an objective function value smaller than f_{opt}. Let $X^{k+1} = X^k \cup Z^k$.

Step 4 (Fathoming) Remove all the integer subboxes in X^{k+1} whose objective function value is larger than f_{opt}.

Step 5 (Optimality Check and Termination) If X^{k+1} is empty, stop and x_{opt} is optimal to (P) with f_{opt} as the objective function value. Otherwise, set $k = k + 1$ and go back to Step 1.

To illustrate the solution process of the proposed convergent surrogate constraint dynamic programming using domain cut, let us consider the following example.

$$\begin{aligned} \min \ & f(x) = 3x_1^2 + 2x_2^2 \\ \text{s.t.} \ & g_1(x) = 2x_1 + 3x_2 \le 7 \\ & g_2(x) = (7 - 2x_1) + (8 - 2x_2) \le 10, \end{aligned}$$

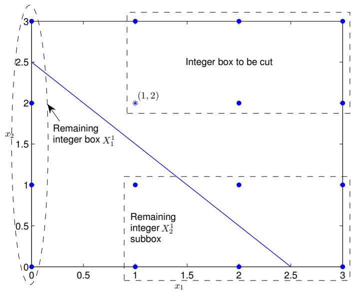

Fig. 9. Domain cut in the 1st iteration

$$x \in X = \left\{x \in \mathbb{Z}^2 \mid 0 \leq x_i \leq 3,\ i = 1, 2\right\}.$$

Applying the surrogate constraint formulation to the above example problem yields,

$$\begin{aligned} &\min\ f(x) = 3x_1^2 + 2x_2^2 \\ &\text{s.t. } \mu_1(2x_1 + 3x_2) + \mu_2(15 - 2x_1 - 2x_2) \leq 7\mu_1 + 10\mu_2, \\ &\qquad x \in X = \left\{x \in \mathbb{Z}^2 \mid 0 \leq x_i \leq 3,\ i = 1, 2\right\}. \end{aligned}$$

Setting $\mu^* = (0, 1)^T$ gives a feasible half-space in x and yields a solution $x^0 = (1, 2)^T$ which violates $g_1(x) \leq 7$. As f is an increasing function, $\langle(1, 2)^T, (3, 3)^T\rangle$ should be removed by case (iv). Let $X^1 = X^0 \setminus \langle(1, 2)^T, (3, 3)^T\rangle = X_1^1 \cup X_2^1 = \langle(0, 0)^T, (0, 3)^T\rangle \cup \langle(1, 0)^T, (3, 1)^T\rangle$. See Figure 9.

Solving (P_{μ^*}) with $\mu^* = (0, 1)^T$ on X_1^1 and X_2^1, respectively, yields a solution on X_1^1, $x_1^1 = (0, 3)^T$, and a solution on X_2^1, $x_2^1 = (2, 1)^T$. Note that $(2, 1)^T$ is feasible with $f(x_2^1) = 14$. Set $(2, 1)^T$ as the incumbent and remove X_2^1 from further consideration. Solution $(0, 3)^T$ violates the first constraint and we cut $\langle(0, 3)^T, (0, 3)^T\rangle$ from X_1^1, resulting

$$X_1^2 = X_1^1 \setminus \langle(0, 3)^T, (0, 3)^T\rangle = \langle(0, 0)^T, (0, 2)^T\rangle$$

Problem (P_{μ^*}) with $\mu^* = (0, 1)^T$ is infeasible on X_1^2 and X_1^2 is removed from further consideration. No more integer box is left and the solution process for the example terminates with the incumbent $(2, 1)^T$ as the optimal solution with $v(P) = 14$.

5. CONCLUSIONS

This paper presents a solid step forward in tackling curse of dimensionality in dynamic programming. Curse of dimensionality vanishes when embedding the primal problem into its surrogate constraint relaxation. However, as a price, dynamic programming needs to be carried out in many sub-domains and in an iterative fashion. Although preliminary computational results are promising, further investigation on many implementation issues is necessary.

REFERENCES

D. P. Bertsekas, and J. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

M. E. Dyer, W. O. Riha and J. Walker. A hybrid dynamic programming/branch-and-bound algorithm for the multiple-choice knapsack problem. *J. Comput. Appl. Math.*, 58:43–54, 1995.

A. O. Esogbue. The status of nonserial dynamic programming. *Manage. Sci.*, 19:350–352, 1972.

F. Glover. A multiphase-dual algorithm for the zero-one integer programming problem. *Oper. Res.*, 13:879–919, 1965.

D. Jacobson, and D. Mayne. *Differential Dynamic Programming.* Elsevier Sci. Publ., New York, 1970.

S. A. Johnson, J. R. Stedinger, and C. A. Shoemaker. Numerical solution of continuous-state dynamic programs using linear and spline interpolation. *Oper. Res.*, 41:484, 1993.

R. E. Larson. Dynamic programming with reduced computational requirements. *IEEE Trans. Automat. Control*, 10:135–143, 1965.

R. E. Larson, and A. J. Korsak. Dynamic programming successive approximations technique with convergence proofs. *Automitica*, 6(2):245, 1970.

D. Li, and Y. Y. Haimes. The envelope approach for multiobjective optimization problems. *IEEE Trans. Sys. Man, Cyber.*, 17(6)1026–1038, 1987.

D. Li, and Y. Y. Haimes. New approach for nonseparable dynamic programming problems. *J. Optim. Theory Appl.* , 64(2):311–330, 1990.

D. Li, and X. L. Sun. *Nonlinear Integer Programming.* Springer, 2006.

L. Z. Liao, and D. Li. Adaptive differential dynamic programming for multiobjective optimal control. *Automatica*, 38:1003–1015, 2002.

L. Z. Liao, and C. A. Shoemaker. Convergence in unconstrained discrete-time differential dynamic programming. *IEEE Trans. Automat. Control*, 36:692–706, 1991.

R. Luus. Iterative dynamic programming: From curiosity to a practical optimization procedure. *Control Intell. Syst.*, 26:1–8, 1998.

F. Korner. A hybrid method for solving nonlinear knsapck problems. *European J. Oper. Res.*, 38:238–241, 1989.

R. E. Marsten, and T. L. Morin. A hybrid approach to discrete mathematical programming. *Math. Program.*, 14:21–40, 1978.

D. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *Int. J. Control*, 3:85–95, 1966.

T. L. Morin, and A. M. O. Esogbue. The imbedded state space approach to reducing dimensionality in dynamic programs of higher dimensions. *J. Math. Anal. Appl.*, 48:801–810, 1974.

S. Yakowitz, and B. Rutherford. Computational aspects of discrete-time optimal control. *Appl. Math. Comput.*, 15:29–45, 1984.