

**EMPLEO DE MODELOS REDUCIDOS Y PROCESAMIENTO DISTRIBUIDO
EN OPTIMIZACIÓN DINÁMICA**

Gustavo E. Vazquez, Nélide B. Brignole

Universidad Nacional del Sur - Avda. Alem 1253 - 8000 Bahía Blanca - Argentina
Planta Piloto de Ingeniería Química (UNS - CONICET)
12 de Octubre 1842 - 8000 Bahía Blanca - Argentina
e-mail: {gvazquez, dybrigno}@criba.edu.ar

RESUMEN

En este trabajo se discuten los alcances y limitaciones de los enfoques secuencial y simultáneo para la solución de problemas de optimización dinámica. Se analizan dos metodologías para reducir el esfuerzo computacional requerido, lo cual permitiría la aplicación más eficiente de estas técnicas a problemas de gran escala. Se propone reducir el modelo dinámico que representa el sistema físico a optimizar y paralelizar los algoritmos de optimización recurriendo a un enfoque "multisplitting".

ABSTRACT

The scope of both the sequential and simultaneous approaches to solve dynamic optimization problems is discussed in this paper. Two strategies to diminish the computational effort involved in these calculations are proposed and analyzed. The run-time decrease would allow the efficient application of these techniques to solve problems of huge size. The proposal consists in using a reduced dynamic model to represent the process under consideration, together with the parallelization of the optimization algorithms by means of a multisplitting approach.

INTRODUCCIÓN

El arranque y parada de equipos, diseño y operación de reactores, diseño de controladores y control óptimo son algunos de los problemas que se modelan a través de sistemas de ecuaciones algebraico-diferenciales, constituyendo lo que se denominan problemas de optimización dinámica.

Una de las principales dificultades que surgen al intentar resolver aplicaciones a sectores de plantas industriales es el tiempo de cómputo asociado. Visto el estado del conocimiento en este tema, surge la necesidad de plantear nuevas alternativas tendientes a disminuir el esfuerzo computacional. Con este propósito, en este trabajo exploramos dos estrategias. La primera se centra en reformular el modelo que representa el sistema a optimizar mientras que la segunda se refiere a las características de los algoritmos. Ambos aspectos, que pueden ser implementados conjuntamente, apuntan a lograr el mismo objetivo, conduciendo a formulaciones más eficientes.

La reformulación del modelo considerada aquí consiste en efectuar una reducción del sistema dinámico, técnica que consiste en transformar el modelo original en otro que contenga menor cantidad de ecuaciones, preservando a su vez las propiedades fundamentales del modelo original.

El segundo enfoque consiste en aumentar el poder de cómputo mediante el uso de algoritmos de optimización paralelos, adoptando fundamentalmente una configuración de procesamiento distribuido heterogéneo: los múltiples procesadores que conforman una computadora paralela son estaciones de trabajo conectadas a una red de transmisión de datos para permitir la comunicación entre ellas. Las ventajas obtenidas por este enfoque son: mínimo costo de inversión (la adaptación de la arquitectura existente para introducir procesamiento distribuido resulta económicamente muy atractiva dado que se aprovechan las estaciones de trabajo disponibles y redes de transmisión de datos estándares), distribución de tareas específicas a estaciones de trabajo apropiadas (en un ambiente de procesamiento distribuido heterogéneo las tareas pueden asignarse a la estación de trabajo más adecuada según sus características) y posibilidades de expansión (el crecimiento del sistema es muy simple, pues sólo implica conectar la nueva estación de trabajo a la red de transmisión de datos). La computación de alta performance (HPC) brinda sistemas poderosos y económicos que constituyen una herramienta ideal para resolver los problemas de optimización dinámica considerados. Por esta razón, su empleo emerge como una necesidad en nuestro medio para que gran parte del software actualmente en desarrollo resulte competitivo.

OPTIMIZACIÓN DINÁMICA

Un problema de optimización dinámica puede formularse como uno de optimización no lineal (NLP) con restricciones algebraicas y diferenciales, de la forma:

$$\begin{aligned} \min_{z(t), y(t), u(t), t_f, p} & \quad \varphi(x(t_f), y(t_f), u(t_f), t_f, p) \\ \text{sujeto a} & \quad F(x(t), \dot{x}(t), y(t), u(t), t, p) = 0 \\ & \quad G(x(t), \dot{x}(t), y(t), u(t), t, p) \leq 0 \end{aligned} \quad (1)$$

donde:

- φ : función objetivo
- F : restricciones de igualdad
- G : restricciones de desigualdad
- x : variables de estado
- y : variables algebraicas

u: variables manipuladas
tf: tiempo final
p: vector de parámetros

La presencia de derivadas de las variables de estado en las restricciones constituye un rasgo distintivo de los problemas de optimización dinámica.

ENFOQUES

Una optimización dinámica se realiza usualmente transformando el problema original en uno de programación no lineal. Este último puede ser resuelto mediante los algoritmos de optimización tradicionales.

Existen dos métodos básicos para efectuar esta transformación. En una de estas técnicas, denominada estrategia simultánea [1], la solución del sistema algebraico-diferencial y el problema de optimización convergen simultáneamente, dado que el modelo dinámico se reformula, convirtiéndolo en ecuaciones algebraicas mediante la técnica de colocación sobre elementos finitos. De este modo, las ecuaciones de colocación pasan a formar parte de las restricciones y los puntos de colocación constituyen variables de decisión adicionales en el nuevo problema. La formulación original trabaja sobre una malla fija. Una mejora tendiente a reducir errores [2] consiste en utilizar una estructura anidada de dos niveles. El lazo interno corresponde al planteo original, mientras que las variables de decisión del lazo externo definen la malla óptima. Gracias a estas discretizaciones es posible establecer restricciones sobre los perfiles de las variables de estado y/o control. Sin embargo, esta transformación provoca un crecimiento exponencial del problema resultante, lo que limita su aplicación a casos de tamaño reducido.

La otra técnica de transformación es el uso de una estrategia secuencial [3]. Asumiendo restricciones de igualdad en (1), la metodología propone aproximar el perfil de control mediante valores discretos a trozos ó por una combinación lineal de funciones pero, a diferencia del enfoque simultáneo, preservando el modelo dinámico. De esta manera, el problema resultante mantiene su dimensión, permitiendo abarcar casos a gran escala; además, el modelo dinámico puede ser usado en su formulación original. Una de las desventajas detectadas es el tiempo de cómputo necesario, dado que en este enfoque debe integrarse el sistema de ecuaciones algebraico-diferencial en cada iteración. Además, debe tenerse en cuenta que sistemas con índice mayor a uno no siempre pueden ser fácilmente integrados. La incorporación de restricciones de desigualdad que involucren variables de estado y/o control no es inmediata. Este aspecto es analizado en [4].

No es posible definir una de estas estrategias como la mejor para transformar un problema de optimización dinámica en NLP. Como hemos discutido arriba, poseen ventajas y limitaciones, razón por la cual la elección correcta surgirá de las características particulares de cada problema.

ALGORITMOS DE OPTIMIZACIÓN PARALELOS

Los algoritmos secuenciales de optimización no lineal tradicionales son extendidos introduciendo el paralelismo sólo en una parte de su carga de cómputo. Esto puede lograrse fácilmente paralelizando la evaluación de la función objetivo y de su gradiente, o mediante el uso de un método paralelo para la solución del sistema algebraico-diferencial [5] cuando se utiliza un enfoque secuencial en el tratamiento del problema de optimización dinámica.

Si bien estas ideas permiten mejorar el desempeño de un algoritmo secuencial clásico en un ambiente de múltiple poder de procesamiento, el paradigma paralelo para la solución de un problema exige la consideración de diversos factores (arquitectura, "overhead" de comunicación de datos, etc.) que sólo pueden ser abordados en un nuevo planteo.

Escasos algoritmos paralelos han sido considerados en la literatura para estos problemas. Entre ellos, [6] propone una búsqueda multidireccional basada en el método directo simplex, fácilmente adaptable a entornos de procesamiento distribuido. Sin embargo, este algoritmo no satisface las necesidades de eficiencia en el tiempo de cómputo, debido a la gran cantidad de evaluación de funciones que realiza.

Un algoritmo paralelo de optimización basado en programación cuadrática sucesiva (SQP) es presentado en [7], pero la granularidad media producida por el acoplamiento interno del algoritmo hace que no se adecue a las necesidades deseadas de implementación en entornos de procesamiento distribuido.

Finalmente, la estrategia de multisplitting [8] parece la más indicada para la clase de problemas considerados en este trabajo. En este caso, la solución de un problema de optimización implica resolver iterativamente en paralelo subproblemas de menor dimensión.

Una estrategia multisplitting en problemas de optimización

Consideremos el problema de optimización no lineal sin restricciones (2), D un entorno de x^* (un punto estacionario único de $f(x)$), y la matriz del Hessiano $G = \nabla^2 f(x)$ semidefinida positiva.

$$\min_{x \in D} f(x) \quad (2)$$

Definición: Multisplitting (MS)

Sea $f_j(x, Y_j) = f(A_j)$, $f_j: \mathbb{R}^n \times \mathbb{R}^{m_j} \rightarrow \mathbb{R}$, $j = 1 \dots p$, x particionado en bloques $x = (X_1, X_2, \dots, X_p)$, y A_j la partición variable de x con el componente j -ésimo reemplazado por Y_j . Entonces decimos que las funciones f_j forman un multisplitting de f y el punto estacionario es definido iterativamente como:

$$x^{k+1} = \sum_{j=1}^p \alpha_j^{k+1} Z^{j,k+1} \quad (3)$$

donde $Z^{j,k}$ es x_k con Y_j en la j -ésima posición e Y_j se obtiene del subproblema:

$$\min_{x_j \in \mathbb{R}^n} f_j(x) \quad 1 \leq j \leq p \quad (4)$$

Nótese que cada subproblema es de dimensión j . Los α_j^k son escalares positivos que satisfacen $\sum_{j=1,p} \alpha_j^k = 1$. El Listado 1 muestra el algoritmo de optimización multisplitting.

```

Para todos los procesadores  $i$ :
|    $k \leftarrow 0$ ;
|   inicializar  $X_i^k$ ;
|   mientras (no converge)
|       |   determinar  $X_j^k \quad j \neq i$ ;
|       |    $Y_i^{k+1} \leftarrow \min f_j(x_k, Y_j)$ ;
|       |   actualizar:  $X_i^{k+1} \leftarrow \alpha_i^{k+1} Y_i^{k+1} - (1 - \alpha_i^{k+1}) X_i^k$ ;
|       |    $k \leftarrow k + 1$ ;
|

```

Listado 1: Algoritmo de optimización multisplitting.

El algoritmo de optimización multisplitting es implementado en forma simple bajo un esquema maestro-esclavo. En cada iteración k , el proceso maestro distribuye en los p procesos esclavos los subproblemas de minimización de los bloques en que se particiona el vector de variables de decisión x . Cuando han sido resueltos, comunican el resultado al proceso maestro para actualizar el problema original. En la práctica es deseable implementar una búsqueda unidimensional sobre los puntos definidos por x_k y x_{k-1} para acelerar la convergencia.

La extensión de esta técnica a problemas de optimización con restricciones puede encontrarse en [8].

OBSERVACIONES Y PROPUESTAS

La técnica de multisplitting posee ventajas altamente considerables. En primer lugar, podemos observar la simplicidad de implementación del algoritmo. No existen restricciones en cuanto a la arquitectura paralela necesaria para su implementación, pues mantiene la característica de granularidad gruesa deseada en procesamiento paralelo distribuido (solo se necesitan $2p$ mensajes en cada iteración) y con un "speed-up" casi lineal con un número considerable de procesadores. Además, la solución de los subproblemas puede realizarse mediante cualquier código de optimización disponible, desde un algoritmo de búsqueda directa hasta un eficiente SQP. En particular, si consideramos la estrategia secuencial para el tratamiento de problemas de optimización dinámica, es deseable implementar un optimizador que requiera menor cantidad de evaluaciones de la función, ya que en cada una de ellas debe integrarse el modelo dinámico asociado.

Sin embargo, podemos notar algunos aspectos no reportados hasta el momento. Asumamos la partición $x = (X_1, X_2, \dots, X_{p^*})^T$, $x \in \mathfrak{R}^n$, y un conjunto de procesadores homogéneo (es decir, con el mismo poder de cálculo) $X_i \in \mathfrak{R}^{n/p^*}$. Esta partición, mantenida en cada una de las iteraciones de algoritmo, esta forzando un desacople en las variables de f que en realidad no existe. Esto provoca que disminuya la "calidad" de los puntos obtenidos en cada iteración, generando una mayor cantidad de iteraciones, de tiempo de comunicación entre las estaciones de trabajo del entorno distribuido, y de tiempo de cómputo.

En este trabajo proponemos realizar un estudio preliminar de sensibilidad de la función objetivo para estimar el acoplamiento de las variables del problema. En un caso ideal podría llegar a obtenerse p^* conjuntos desacoplados de igual dimensión, de manera que la partición establecida es óptima e inamovible durante la ejecución del algoritmo. Naturalmente, la realidad indica que la cantidad de particiones y las dimensiones de las mismas están alejadas del ideal p^* y n/p^* respectivamente. Si el estudio realizado indica particiones óptimas de tamaño mayor a n/p^* nuevamente imponemos un desacople ficticio. Supongamos el caso de una acople Q que involucra q variables, y que por su tamaño deba ser dividido en w particiones. Una solución simple y efectiva es considerar a las particiones w_i con elementos distintos de Q en cada iteración. De esta manera se evita forzar un desacople ficticio siempre sobre las mismas variables. Las particiones óptimas de tamaño menor a n/p^* no generan inconvenientes y pueden ubicarse convenientemente.

Las pruebas realizadas sobre casos de estudio tales como la función de Wood y los ejemplos reportados en [8] indican prometedores resultados con reducciones de un 30-40% en la cantidad de iteraciones y el tiempo de cómputo frente a la formulación original de multisplitting. En el caso de no realizar un análisis del acople de las variables del problema, la heurística consistente en variar los elementos que conforman cada partición en las iteraciones también conduce a buenos resultados.

En cuanto al tamaño de cada partición vale hacer algunas consideraciones. Es de esperar que minimizar una partición con sus elementos acoplados consuma más tiempo de cómputo que otra de igual dimensión con sus elementos desacoplados. El análisis de sensibilidad puede ser usado para estimar la complejidad de cálculo de una partición. De esta manera, la dimensión de las particiones puede variarse dinámicamente (en tiempo de ejecución) para disminuir el tiempo muerto de procesamiento en cada iteración. Un tratamiento similar es válido para el caso de un entorno distribuido heterogéneo (es decir, estaciones de trabajo con diferente capacidad de cálculo), de manera que sea posible distribuir subproblemas de distinta dimensión según el poder de cómputo de cada procesador.

REDUCCIÓN DE MODELOS

Existen dos procedimientos básicos para disminuir el esfuerzo computacional asociado a un sistema de ODEs a nivel de modelamiento. Por un lado, la simplificación de modelos implica descartar algunas de sus características originales, reduciendo la complejidad y/o cantidad de ecuaciones del mismo. Por otro, la reducción intenta transformar el modelo original en otro que contenga menor cantidad de ecuaciones, aunque preservando las propiedades del modelo

¹ Podemos considerar p^* como la óptima cantidad de particiones determinada por el máximo speed-up del problema, o siguiendo consideraciones prácticas o de hardware disponible.

original. Una técnica de reducción ideal debería satisfacer, desde el punto de vista de la matemática aplicada, los requerimientos de precisión (buena concordancia entre los resultados del modelo riguroso y el reducido), robustez (poder resolver un amplio espectro de problemas) y eficiencia (rapidez de cálculo con bajo costo en el almacenamiento de los datos). Además, desde el punto de vista de las aplicaciones concretas, debe preservar la claridad de la relación entre el modelo y el proceso físico representado, mantener el significado físico de las variables y al menos preservar la cantidad de parámetros y el grado de dificultad para ajustarlos. En muchos casos, la rigurosidad debe ser preservada en el modelo de bajo orden; por lo tanto, la técnica de reducción se presenta como más indicada.

Las dos estrategias para transformar problemas de optimización dinámica en NLP dependen fuertemente del tamaño del modelo dinámico. Es aconsejable reducir el modelo antes de efectuar la transformación. En la estrategia secuencial, disponer de un buen modelo reducido resulta conveniente, pues el sistema de ecuaciones diferenciales debe ser integrado repetidamente, ocasionando un tiempo de cómputo prohibitivo para sistemas muy grandes. En la estrategia simultánea, aproximar el modelo dinámico reducido mediante colocación permite disminuir considerablemente el tamaño del problema NLP resultante, pues cada ecuación diferencial eliminada implica la desaparición de varias variables y restricciones algebraicas en el problema NLP resultante.

A continuación describimos brevemente algunas metodologías para efectuar reducciones de modelos.

- Técnica de Autovalores Dominantes [9],[10]

Esta metodología consiste en particionar el sistema original (linealizado) en un subsistema diferencial que refleja los modos dominantes y un subsistema algebraico con los restantes estados, cuantificando el efecto de cada autovalor de la matriz A sobre todas las salidas y , identificándose los modos dominantes. Este objetivo puede alcanzarse representando solamente los fenómenos lentos asociados a las respuestas. La técnica original preserva una formalización matemática, aunque se limita a sistemas de ODEs lineales invariantes en el tiempo.

- Técnica de compartimentación [11]

La técnica de compartimentación propone dividir el conjunto de ODEs en subconjuntos de ecuaciones cuyas soluciones tienen una forma similar. De esta manera, se asocia la respuesta correspondiente a cada subconjunto de ecuaciones (compartimiento) a una de sus variables denominada sensible. Las variables restantes que forman el compartimiento se calculan a través de relaciones algebraicas. Así, el vector de variables diferenciales puede permutarse para lograr la partición $x = [X_s \ X_{ns}]^T$, donde X_s contiene todas las variables sensibles. El modelo reducido se define como:

$$\frac{dX_s}{dt} = \tilde{f}(X_s, u, t) \qquad X_{ns} = \tilde{g}(x, u, t)$$

La calidad del modelo reducido resultante depende de la forma en que se eligen los compartimientos, sus correspondientes variables sensibles y el cálculo de las variables no

sensibles. Tanto este trabajo como algunas variantes propuestas por otros autores ([12],[13]) proponen políticas de selección fuertemente dependientes de la aplicación específica, sin formalizarlas rigurosamente.

- Técnica de compartimentación basada en autovalores dominantes [14]

En esta técnica se propone usar la noción de dominancia de autovalores para la elección de las variables sensibles en el modelo reducido y una nueva formulación del sistema de ecuaciones algebraicas para el cálculo de las variables no sensibles. La formalización matemática presentada en este trabajo para determinar estos aspectos introduce sensibles mejoras a la técnica de compartimentación original, ya que se logra mucha menor discrepancia entre los valores de las variables de estado calculados mediante los modelos riguroso y reducido.

- Técnica de colocación ortogonal [15]

La reducción de modelos a partir de la técnica de colocación ortogonal se obtiene aproximando las variables de estado por una función de interpolación. La elección de estos puntos da lugar a distintas variantes del método. Dado que los puntos de colocación se eligen en base a consideraciones matemáticas, las variables de estado del modelo reducido no necesariamente poseen un sentido físico. Esto constituye una desventaja y puede conducir a discrepancias entre los estados estacionarios correspondientes al modelo completo y reducido.

CONCLUSIONES

El estudio realizado nos permitió definir la mejor política para la resolución de problemas de optimización dinámica. La propuesta consiste en usar la estrategia multisplitting para paralelizar el algoritmo de optimización trabajando en un entorno de procesamiento distribuido. En este sentido, proponemos una variante para efectuar el particionamiento del vector de variables de decisión. Esto permite reducir considerablemente la cantidad de iteraciones requeridas. Conjuntamente, se sugiere emplear un modelo dinámico con la mínima cantidad posible de ecuaciones diferenciales. Para lograr este objetivo se recomienda aplicar la técnica de reducción por compartimentación basada en dominancia de autovalores, que permite preservar apropiadamente la calidad de las respuestas dinámicas del modelo original.

REFERENCIAS

- [1] Cuthrell J.E. and L.T. Biegler, *On the optimization of differential-algebraic process systems*. AIChE J 33, 1257-1270, 1987.
- [2] Tanarkit P. and L.T. Biegler, *A nested, simultaneous approach for dynamic optimization problems-II: the outer problem*. Comunicación personal, 1996.
- [3] Vassiliadis, V., *Computational solution of dynamic optimization problems with general differential-algebraic constraints*. Ph.D. Dissertation, University of London, London, UK, 1993.

-
- [4] Feehery W.F. and P.I. Burton, *Dynamic simulation and optimization with inequality path constraints*. Comp. Chem. Engng. 20, S707-S712, 1996.
 - [5] Bahi J., E. Griepentrog and J.C. Miellou, *Parallel treatment of a class of differential-algebraic systems*. Siam Journal on Numerical Anal. 33, no.5, 1969-1980, 1996.
 - [6] Dennis J. and V. Torczon, *Direct search methods on parallel machines*. Siam Journal on Optimization 1, 937-954, 1991.
 - [7] Ghattas O. and Orozco C.E., *A parallel reduced hessian SQP method for shape optimization*. Technical report, Computational Mechanics Lab, Carnegie-Mellon University, 1996.
 - [8] Renaut R.A. and H.D. Mittelmann, *Parallel multisplitting for optimization*. Technical report, Department of Mathematics, Arizona State University, 1994.
 - [9] Litz L., *Order Reduction of linear state-space models via optimal approximation of the nondominant modes*. Proc. IFAC, 195-202, 1980.
 - [10] Litz L. and H. Roth, *State decomposition for singular perturbation order reduction-A modal approach*. Int. J. Control 34, no.5, 937-954, 1981.
 - [11] Benallou A., D. E. Seborg and D. A. Mellichamp, *Dynamic compartmental models for separation processes*. AIChE J 32, 1067-1078, 1986.
 - [12] Horton R. R., B. W. Bequette and T. F. Edgar, *Improvements in dynamic compartmental modeling for distillation*. Comp. Chem. Engng. 15, 197-201, 1991.
 - [13] Tolliver T.L. and L.C. McCune, *Distillation control design based on steady state simulation*. ISA Trans. 17, 3-10, 1978.
 - [14] Vazquez G.E., Ugrin P.E., Brignole N.B. *Evaluación de Técnicas de Reducción de Sistemas de Ecuaciones Diferenciales Ordinarias*. Mecánica Computacional, Vol. XVI, 361-371, 1996.
 - [15] Cho Y. S. and B. Joseph, *Reduced-order steady-state and dynamic models for separation processes*, AIChE J 29, 261-276, 1983.

