lsmear: A Variable Selection Strategy for Interval Branch and Bound Solvers

Ignacio Araya, Bertrand Neveu

`lsmear`: A Variable Selection Strategy for Interval Branch and Bound Solvers

Ignacio Araya[1] and Bertrand Neveu[2]

[1] *Pontificia Universidad Católica de Valparaíso, Escuela de Ingeniería Informática, Chile,* ignacio.araya@pucv.cl

[2] *Imagine LIGM Université Paris–Est, France,* Bertrand.Neveu@enpc.fr

Abstract Smear-based variable selection strategies are well-known and commonly used by branch-and-prune interval-based solvers. They estimates the impact of the variables on each constraint of the system by using the partial derivatives and the sizes of the variable domains. Then they aggregate these values, in some way, to estimate the impact of each variable on the whole system. The variable with the greatest impact is then selected. A problem of these strategies is that they, generally, consider all constraints equally important.

In this work, we propose a new variable selection strategy which first weights the constraints by using the optimal Lagrangian multipliers of a linearization of the original problem. Then, the impact of the variables is computed with a typical smear-based function but taking into account the weights of the constraints. The strategy is tested on classical benchmark instances outperforming significantly the classical ones.

Keywords: global optimization, branch and bound, variable selection, Lagrangian multipliers, smear function

1. Introduction

This paper deals with continuous global optimization (nonlinear programming) deterministically handled by interval branch and bound (B&B). The problem is defined by: $\min_{x \in \boldsymbol{x}} f(x) \;\; s.t. \;\; g(x) \leq 0$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the real-valued objective (non convex) function and $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a vector-valued (non convex) function.[1] $x = (x_1, ..., x_i, ...x_n)$ is a vector of variables varying in a domain (i.e., a box) $\boldsymbol{x}$[2]. For performance and simplicity considerations, a variable x_o, with initial domain $\boldsymbol{x}_o = [-\infty, +\infty]$ is included in the set of variables x and an additional constraint $f(x) = x_o$ is included in the set of constraints (actually, functions $f(x) - x_o$ and $x_o - f(x)$ are included in g). Finally, we solve an equivalent problem:

$$\min_{x \in \boldsymbol{x}} x_o \;\; s.t. \;\; g(x) \leq 0 \tag{1}$$

Several works have been proposed for finding good branching strategies ([2, 4, 7, 5, 3]). Smear-based methods [4, 3] use information on the system to obtain the variable with the greatest *impact*. The impact of a variable x_i on a function g_j is computed by means of the *smear value*. Consider that the current node is associated with box $\boldsymbol{x}$; the smear value is given by: $\text{smear}(x_i, g_j) = |\boldsymbol{J}_{ji}| * \text{wid}(\boldsymbol{x}_i)$, where $\boldsymbol{J}_{ji}$ is an interval overestimate of the range of the partial derivative $\frac{\partial g_j}{\partial x_i}$ in $\boldsymbol{x}$. $|\boldsymbol{J}_{ji}|$ is the *magnitude* of the interval $\boldsymbol{J}_{ji}$, i.e., $|\boldsymbol{J}_{ji}| = \max(|\underline{J_{ji}}|, |\overline{J_{ji}}|)$.

[1] The branching strategies proposed in this paper can also apply to problems having equality constraints.

[2] An interval $\boldsymbol{x}_i = [\underline{x_i}, \overline{x_i}]$ defines the set of reals x_i s.t. $\underline{x_i} \leq x_i \leq \overline{x_i}$. A *box* $\boldsymbol{x}$ is a Cartesian product of intervals $\boldsymbol{x}_1 \times ... \times \boldsymbol{x}_i \times ... \times \boldsymbol{x}_n$.

Selection methods based on the smear value select the variable that maximizes an aggregation of this value in the whole system.

Tawarmalani and Sahinidis [6] present an algorithm called *ViolationTransfer*, to estimate the impact of a variable on the problem. ViolationTransfer works with the Lagrangian function of a relaxation of the problem and an optimal solution of the relaxation $x^\star$. For each variable, an interval $\boldsymbol{x}_i^v \subset \boldsymbol{x}_i$ is defined. $\boldsymbol{x}^v$ is the smallest box such that it contains $x^\star$ and each univariate constraint $g_j(x_1^\star, ..., x_{i-1}^\star, x_i, x_{i+1}^\star, x_n^\star) \leq 0$ $(j = 1..m)$ is feasible for at least one value in $\boldsymbol{x}_i^v$.

Then, for each variable x_i, the difference between the bounds of the image of the Lagrangian function over the interval $\boldsymbol{x}_i^v$ is estimated. In each estimation all the variables are fixed except x_i. The assumption is that branching on the variable maximizing the image width is likely to improve the lower bound of the objective function in the subproblems.

In this article we propose `lsmear`, a new variable selection strategy for interval B&B solvers. In a few words, the method selects the variable maximizing the smear value of the Lagrangian function of the problem. In the Lagrangian function, the Lagrange multipliers are replaced by the dual optimal of a linear approximation of the problem. Related to the ViolationTransfer strategy our approach has some important differences:

1. `lsmear` uses a simple *linear approximation* of the original problem instead of sophisticated convex relaxation techniques.

2. `lsmear` estimates the impact of each variable in the Lagrangian function of the *original problem*. The estimated impact is computed by using the smear value.

3. The computation of $\boldsymbol{x}^v$ requires the solver uses a reformulated problem in which multidimensional functions are replaced with either univariate or bilinear functions [6]. For the moment, and in order to maintain the simplicity and generality of the approach, `lsmear` uses directly $\boldsymbol{x}$ instead of $\boldsymbol{x}^v$.

2. `lsmear`, a Smear-based strategy using optimal Lagrange multipliers

A main issue related to the smear-based strategies is that these strategies consider all the constraints equally important. To overcome this issue we propose to estimate the impact of the constraints in the system by using the optimal Lagrange multipliers of a linear approximation of the original problem.

The `lsmear` method works in two phases. First, a linearization of the global optimization problem is generated. Each function $g_j(x)$ is approximated by using the first order term of its Taylor expansion around the midpoint of the box, i.e.,

$$gl_j(x) = g_j(\text{mid}(\boldsymbol{x})) + \sum_{i=1}^{n} \text{mid}(\boldsymbol{J}_{ji}).(x_i - \text{mid}(\boldsymbol{x}_i))$$

where $\boldsymbol{J}_{ji}$ is an interval overestimation of the image of $\frac{\partial g_j}{\partial x_i}$ over $\boldsymbol{x}$. Note that instead of using the partial derivatives in the midpoint of the box we use the midpoint of the overestimation of the partial derivatives (i.e., $\text{mid}(\boldsymbol{J}_{ji})$).

The generated linear optimization problem includes the bound constraints, i.e., $\underline{x_i} \leq \boldsymbol{x}_i \leq \overline{x_i}$, and it is solved by using the simplex method[3]. If an optimum exists, then a second phase is carried out. In this phase, the strategy computes the smear value of the following function:

$$L(x) = x_o + \sum_{i=1}^{m} \lambda_j^* g_j(x)$$

[3] Actually we need to solve the dual problem, however we use a linear solver which solves both, the primal and the dual problems

where λ^* corresponds to the dual optimal solution of the linear problem. The function L is equivalent to the Lagrangian of (1) but the Lagrange multipliers have been replaced by the optimal Lagrange multipliers of the linear approximation. Thus, the problem of minimizing $L(x)$ can be seen as a rough approximation of the original optimization problem. The interesting thing about L is that it aggregates the objective function and the constraints of the problem in only one function, thus, computing the smear value of each variable in L offers an estimation of the impact of the variable in the original problem. We believe that this estimation is fairer than the one computed by the classical smear-based strategies because each constraint is, in a certain way, being pondered according to its influence on the optimal value. Finally, same as the other Smear-based strategies, the variable with the greater impact is selected for bisection.

Algorithm 1 `lsmear`

```
procedure lsmear($\boldsymbol{x}, \boldsymbol{J}, g$); out: var
    /* Phase 1: linearization and solving the linear program*/
    $gl \leftarrow g(\text{mid}(\boldsymbol{x})) + \text{mid}(\boldsymbol{J}).(x - \text{mid}(\boldsymbol{x}))$
    $\boldsymbol{t} \leftarrow x_o$; $\boldsymbol{x}_o \leftarrow [-\infty, \infty]$
    $\lambda^* \leftarrow$ optimize($\min x_o$, subject to: $\underline{x_i} \leq x_i \leq \overline{x_i},\ gl_j(x) \leq 0$)
    $\boldsymbol{x}_o \leftarrow \boldsymbol{t}$ if $\lambda^* \neq \varnothing$ then
        /* Phase 2: computing the impact of $L(x)$ */
        for $i \in \{1..n\}$ do
            $\boldsymbol{D} \leftarrow \lambda^*_i$ for $j \in \{1..m\}$ do
                $\boldsymbol{D} \leftarrow \boldsymbol{D} + \lambda^*_{n+j}.\boldsymbol{J}_{ji}$
            end
            $I \leftarrow |\boldsymbol{D}.\text{wid}(\boldsymbol{x}_i)|$
            if $I >$ max_impact then
                max_impact $\leftarrow |\boldsymbol{I}|$
                var $\leftarrow i$
            end
        end
        return var
    else
        return smearsum($\boldsymbol{x}, \boldsymbol{J}$)
    end
end.
```

Algorithm 1 shows our approach. $\boldsymbol{J}$ corresponds to the Jacobian matrix which contains the interval overestimations of the partial derivatives over $\boldsymbol{x}$. In the linear program (Phase 1), the interval related to the objective variable is unbounded to enhance the chances for successful finding an optimal solution. In Phase 2, for each variable x_i we first compute $\boldsymbol{D}$, which is an interval overestimation of $\frac{\partial L}{\partial x_i}$ over $\boldsymbol{x}$. The overestimation is obtained by adding the products of the interval partial derivatives on each constraint ($\boldsymbol{J}_{ji}$) and the corresponding dual optimal value (λ^*_{n+j}). $\boldsymbol{D}$ is initialized with the dual optimal value related to the bounded constraint, i.e., λ^*_i (the partial derivative related to the i-th bound constraint over the variable x_i is 1). Then, the smear impact of the variable is computed as the magnitude of the product of the interval partial derivative and the width of the related interval. The variable with the maximum impact is saved and returned. If the linear program does not have solutions or if the optimal value is unbounded, then the `smearsum` method is launched instead. This method uses the same Jacobian matrix received by the `lsmear` one.

3. Experiments

In order to validate our approach, we implemented `lsmear` in IbexOpt, a state-of-the-art optimizer of the Interval-Based EXplorer library (Ibex ([1])). All the experiments were run on a server PowerEdge T420, with 2 quad-processors Intel Xeon, 2.20 GHz and 8 GB RAM.

The instances were selected from the series 1 and 2 of the COCONUT constrained global optimization benchmarks.[4] We selected all the problems solved by some strategy in a time comprised between 2s and 3600s (66 instances). Each strategy was run 5 times on each instance and the average CPU time was reported.

We compared the proposed strategy `lsmear` with some of the classical variable selection strategies: `round-robin` (rr), `largest-first` (lf), `smearsum` (ssum), `smearmax` (smax) and `smearsumrel` (ssr). Figure 1 summarizes the comparison among the six strategies.

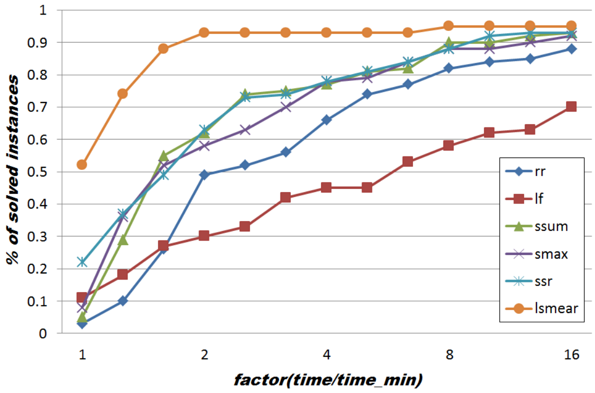

Figure 1. Performance profile.

Each curve reports the percentage of instances solved by the corresponding strategy in less than *factor* times the best reported CPU time. From the results we observe that `lsmear` clearly outperforms all the classical variable selection strategies. Also note that more than 90% of the instances are solved by `lsmear` in less than twice the best CPU time reported by all the strategies.

References

[1] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[2] Tibor Csendes and Dietmar Ratz. Subdivision direction selection in interval methods for global optimization. *SIAM Journal on Numerical Analysis*, 34(3):922–938, 1997.

[3] Eldon Hansen and G William Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.

[4] R Baker Kearfott and Manuel Novoa III. Algorithm 681: INTBIS, a portable interval Newton/bisection package. *ACM Transactions on Mathematical Software (TOMS)*, 16(2):152–157, 1990.

[5] R Moore. Interval analysis, vol. 60, 1966.

[6] Mohit Tawarmalani and Nikolaos V Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3):563–591, 2004.

[7] G. Trombettoni, I. Araya, B. Neveu, and G. Chabert. Inner regions and interval linearizations for global optimization. In *AAAI*, pages 99–104, 2011.

[4] `www.mat.univie.ac.at/~neum/glopt/coconut/Benchmark/Benchmark.html`