A heuristic for nonlinear global optimization relevant to discrete choice models estimation

Michaël Thémans   Nicolas Zufferey
Michel Bierlaire

Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne

TRANSP-OR

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Outline

- Optimization problem formulation
- Motivation: Discrete choice models estimation
  - Objective function highly nonlinear and noncave
  - Several local optima
- Nonlinear global optimization
  - Literature review
  - Different approach
- Features of our new algorithm
- Numerical results
- Conclusions and perspectives

Optimization problem formulation

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$

- is twice differentiable
- is nonlinear and nonconcave
- may present several (and possibly many) local minima
- is usually expensive to evaluate

Nonlinear optimization

- Vast literature in nonlinear optimization
- Drawback: most of the methods and sofwares can only ensure to converge to a local minimum
- Convergence toward a global one cannot be guaranteed
- Several transportation applications require a global minimum of the related optimization problems such as
  - traffic equilibrium problems
  - discrete choice models estimation

$\Rightarrow$ Nonlinear global optimization

Discrete choice models

- Only the Multinomial Logit model (MNL) and the Nested Logit model (NL) are quite easy to estimate
- The corresponding log-likelihood function is globally concave for the MNL and concave in a subset of the parameters for the NL
- Other GEV and mixtures of GEV are more problematic as several difficulties may arise
  - The log-likelihood function and its derivatives become expensive to compute and highly nonlinear and nonconcave
  - Overspecification issues
  - Non trivial constraints on parameters
- Existing softwares can only provide a local optimum of the maximum likelihood estimation problem

Nonlinear global optimization

Most of the deterministic approaches can be grouped in 4 categories

- methods based on real algebraic geometry
- exact algorithms (such as Branch & Bound)
- interval analysis
- difference of convex functions programming (DC)

Nonlinear global optimization

The use of heuristics to address this problem in practice is intensive

- Continuous adaptations of heuristics from discrete optimization
  - Simulated Annealing (Locatelli, 2000)
  - Genetic Algorithms (Chelouah and Siarry, 2000)
  - Tabu Search (Chelouah and Siarry, 2003)
- Hybridation of heuristics with derivative-free methods or random searches
  - Simulated Annealing hybridized with Nelder-Mead algorithm (Hedar and Fukushima, 2002) or with approximate descent direction and pattern search (Hedar and Fukushima, 2004)
  - Tabu search hybridized with direct search methods (Hedar and Fukushima, 2006)
  - Particle swarm heuristic with pattern search (Vaz and Vicente, 2007)

New approach

- Variable Neighborhood Search (VNS) in order to diversify and explore
- 2 essential elements in the VNS
  - Local search used
  - Definition of neighborhoods and neighbors
- Use at best the information on the objective function and its derivatives obtained at best cost
- Limit the number of evaluations of f by identifying promising areas of search and those which are not interesting

VNS for continuous optimization

- Select a set of neighborhoods $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$ and an initial solution x_*^0 (local minimum)
- $x_c = x_*^0$ and $x_*^{best} = x_*^0$
- $k = 1$
- While $k \leq k_{max}$
  - Generate p neighbors of x_c in $\mathcal{N}_k$
  - Apply k_2 iterations of the local search
  - Apply the local search to the best point obtained previously in order to get a local minimum x_*^{new}
  - If $f(x_*^{new}) < f(x_*^{best})$, $x_*^{best} = x_c = x_*^{new}$ and $k = 1$. Otherwise $k = k + 1$.
- The solution (hopefully a global minimum) is x_*^{best}

Local search

- Trust-region method
- Iterative method for unconstrained nonlinear optimization
- Able to efficiently identify a local minimum of the problem
- Globalization technique: convergence from remote starting points (not only in a neighborhood of a local minimum)
- Use of an approximate quadratic model of f
  - ∇f approximated using finite differences
  - $\nabla^2 f$ approximated using the SR1 secant method (quasi-Newton)

Promising areas of search

- The local search is efficient but cumbersome
- Prevent the algorithm from blindly applying the local search and limit the number of evaluations of f
- Perform tests on each iterate x_k generated by the local search
- Given $X_* = \{x_*^0, x_*^1, \ldots\}$ and f_{min}, we define 3 conditions
  - $\exists\ i$ such that $\|x_k - x_*^i\| \leq \varepsilon_1$
  - $\|\nabla f(x_k)\| \leq \varepsilon_2$ and $f(x_k) - f_{min} \geq \varepsilon_3$
  - $f(x_k) > f(x_{k-1}) + \beta \nabla f(x_{k-1})^T s_{k-1}$ and $f(x_k) - f_{min} \geq \varepsilon_3$
- If one of these conditions is verified, we prematurely interrupt the local search

Neighborhoods of the VNS

- The neighborhood $\mathcal{N}_k$ is defined by a distance d_k
- Analysis of the eigen-structure of $\nabla^2 f(x_c) \in \mathbb{R}^{n \times n}$: computation of the n eigen vectors q_i and associated eigenvalues c_i (curvature)
- $2n$ possible directions to determine a neighbor
- Idea: Prefer directions associated with high curvature but...
  - Local information, around x_c
  - Reduce the importance of curvature in the neighbors selection for large neighborhoods

Neighborhoods of the VNS

- Logit-like probability for the selection of a direction:

$$P(q_i) = P(-q_i) = \frac{e^{\lambda \frac{c_i}{d_k}}}{2 \sum_{j=1}^{n} e^{\lambda \frac{c_i}{d_k}}}$$

  where λ is a weight factor associated with curvature

- p directions are randomly selected at each phase of the VNS accordingly to the probability vector P
- If q_i is chosen, the associated neighbor is given by:

$$x_c + \alpha * d_k * q_i$$

  where $\alpha \in U([0.75, 1])$

New VNS

- Select a set of neighborhoods $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$ and an initial solution x_*^0 (local minimum)
- $x_c = x_*^0$ and $x_*^{best} = x_*^0$
- $k = 1$
- While the stopping criterion is not satisfied
  - Generate p neighbors of x_c in $\mathcal{N}_k$
  - Apply the local search from each neighbor while a promising area is detected
  - If all local searches have been prematurely interrupted, $k = k + 1$
  - Otherwise x_*^{new} is the best found local minimum. If $f(x_*^{new}) < f(x_*^{best})$, $x_*^{best} = x_c = x_*^{new}$ and $k = 1$. Otherwise $k = k + 1$.
- The solution (hopefully a global minimum) is x_*^{best}

Some algorithmic details

- Initialization
  - Generate m random points and apply k_2 iterations of the local search
  - Apply the local search to the best point obtained
- Stopping criterion
  - $k > k_{max}$ (stop after k_{max} unsuccessful VNS phases)
  - Maximum CPU time attained
  - Maximum number of evaluations of f attained

Numerical tests

- 15 classical test functions
- 25 associated optimization problems
- 100 runs for small-sized problems
- 20 runs for large-sized problems
- A run is successful if a global minimum is found
- 2 measures of performance
  - Average percentage of success
  - Average number of function evaluations (across successful runs)

Competitors

- 6 algorithms
  - Our proposed VNS
  - General VNS (GVNS, 2006)
  - Directed Tabu Search (DTS, 2006)
  - Simulated Annealing Heuristic Pattern Search (SAHPS, 2004)
  - Continuous Hybrid Algorithm (CHA, 2003)
  - Direct Search Simulated Annealing (DSSA, 2002)
- Different stopping criterion for GVNS

  $\Rightarrow$ specific comparison of VNS and GVNS in terms of number of evaluations

Average percentage of success

Problem	VNS	CHA	DSSA	DTS	SAHPS	GVNS
RC	100	100	100	100	100	100
ES	100	100	93	82	96	
RT	84	100	100		100	
SH	78	100	94	92	86	100
R_2	100	100	100	100	100	100
Z_2	100	100	100	100	100	
DJ	100	100	100	100	100	
$H_{3,4}$	100	100	100	100	95	100
$S_{4,5}$	100	85	81	75	48	100
$S_{4,7}$	100	85	84	65	57	
$S_{4,10}$	100	85	77	52	48	100
R_5	100	100	100	85	91	
Z_5	100	100	100	100	100	
$H_{6,4}$	100	100	92	83	72	100
R_{10}	100	83	100	85	87	100
Z_{10}	100	100	100	100	100	
HM	100		100			
GR_6	100		90			
GR_{10}	100					100
CV	100		100			
DX	100		100			
MG	100					100
R_{50}	100	79		100		
Z_{50}	100	100		0		
R_{100}	100	72		0		

Average number of evaluations of f

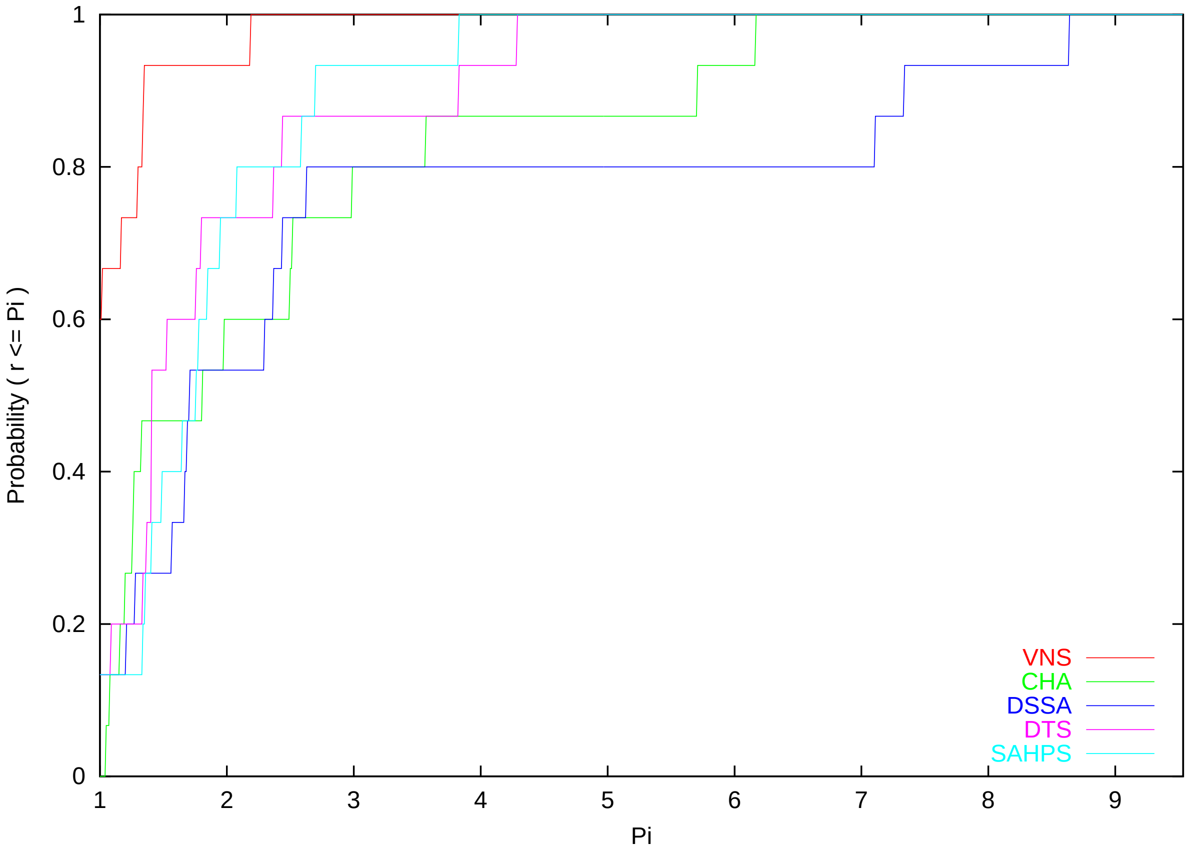
Problem	VNS	CHA	DSSA	DTS	SAHPS
RC	153	295	118	212	318
ES	167	952	1442	223	432
RT	246	132	252		346
SH	366	345	457	274	450
DJ	104	371	273	446	398
$H_{3,4}$	249	492	572	438	517
$H_{6,4}$	735	930	1737	1787	997
$S_{4,5}$	583	698	993	819	1073
$S_{4,7}$	596	620	932	812	1059
$S_{4,10}$	590	635	992	828	1035
R_2	556	459	306	254	357
Z_2	251	215	186	201	276
R_5	1120	3290	2685	1684	1104
Z_5	837	950	914	1003	716
R_{10}	2363	14563	16785	9037	4603
Z_{10}	1705	4291	12501	4032	2284
HM	335		225		
GR_6	807		1830		
CV	854		1592		
DX	2148		6941		
R_{50}	11934	55356		510505	
Z_{50}	17932	75520		177125	
R_{100}	30165	124302		3202879	

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Performance profile

- 15 problems
- Number of evaluations of f

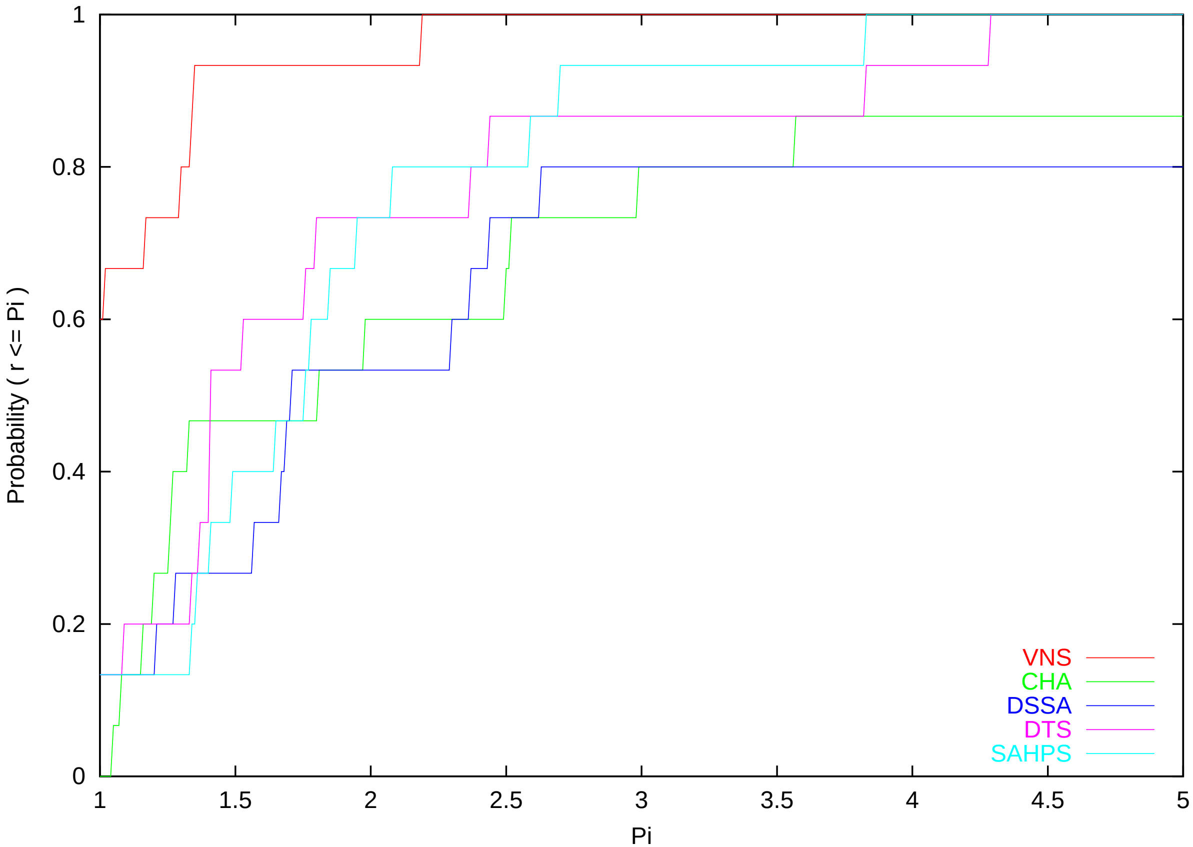


TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Performance profile

- 15 problems
- Number of evaluations of f - zoom on π between 1 and 5



Large size

- CPU time in seconds on problems of sizes 50 and 100

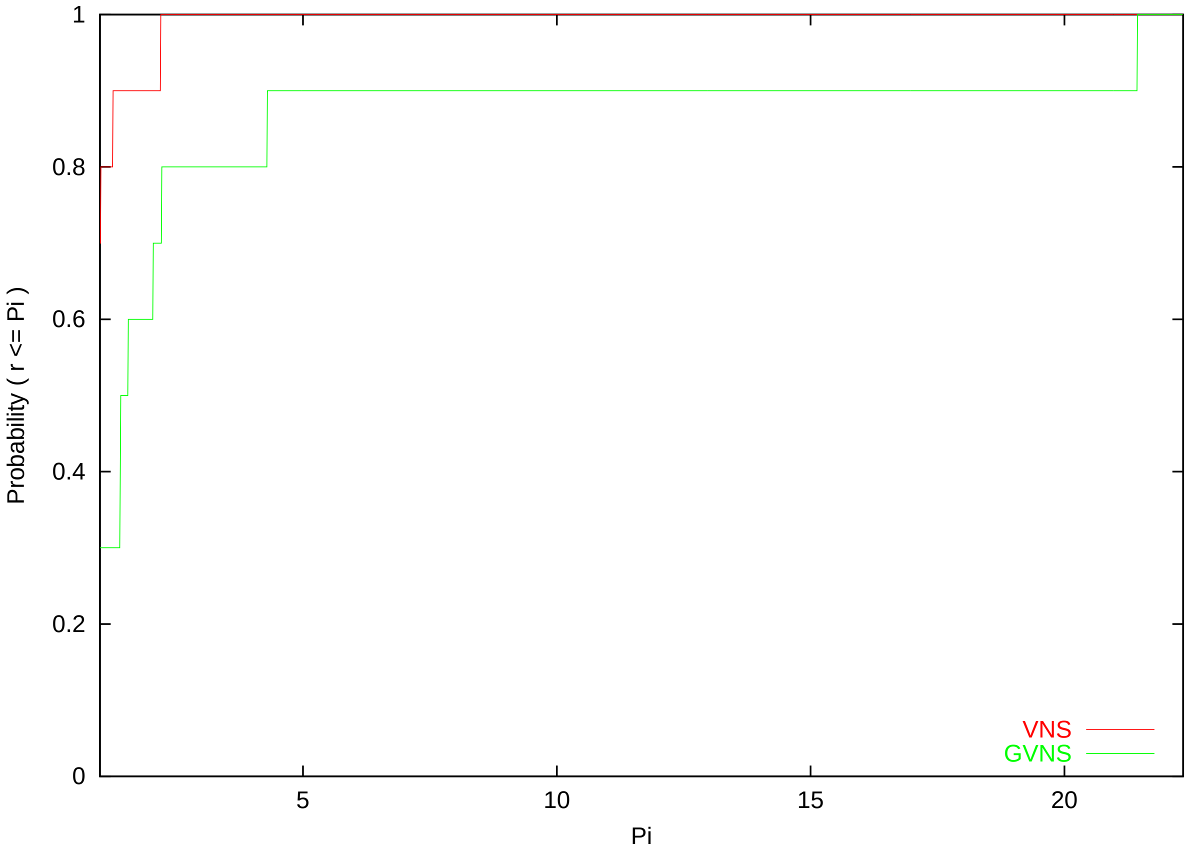
Problem	VNS	DTS
R_{50}	208	1080
Z_{50}	228	1043
R_{100}	1171	15270

Average number of evaluations of f

Problem	VNS	GVNS
RC	99	45
SH	305	623
R_2	176	274
R_{10}	1822	39062
GR_{10}	1320	1304
$H_{3,4}$	174	385
$H_{6,4}$	532	423
$S_{4,5}$	468	652
$S_{4,10}$	481	676
MG	17	73

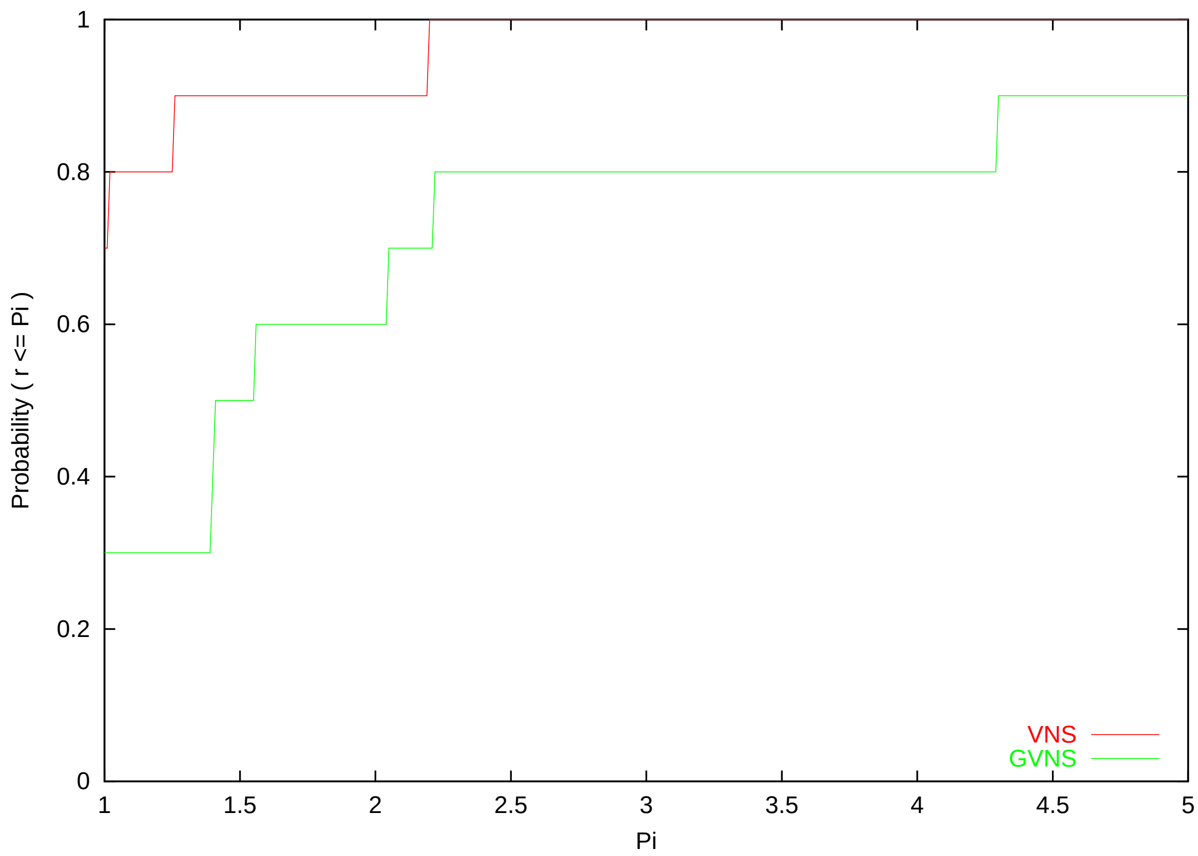
Performance profile

- 10 problems
- Number of evaluations of f



Performance profile

- 10 problems
- Number of evaluations of f - zoom on π between 1 and 5



Conclusions

- Algorithm for nonlinear global optimization relevant to discrete choice models estimation
  - New VNS for continuous optimization
  - Intensive use of information on f
  - Intelligent local search and neighborhoods
  - Collaboration between nonlinear optimization and discrete optimization
- Very good behavior in conducted numerical experiments
  - The VNS allows to diversify and explore: good robustness
  - The computational cost of the local search is compensated by its efficiency and the identification of promising areas

Perspectives

- Additional numerical experiments
  - Other test problems
  - Tests in large size
  - Tests with BIOGEME to estimate discrete choice models
  - Evaluate the performance for a given budget of time or computational cost
- Algorithmic developments
  - Better identification of convergence basins
  - p dynamic
  - Incorporate into an Adaptive Memory Method framework
  - Generalization to constrained optimization

Thank you for your attention !