Artelys Knitro 10.2 : Moindres Carrés Non-Linéaires

Michaël Gabay[1], Jean-Hubert Hours[1]

Artelys S.A., 81 rue Saint-Lazare, 75009 Paris, France

{michael.gabay,jean-hubert.hours}@artelys.com

Mots-clés : *solveur, optimisation non-linéaire, optimisation en nombres entiers, optimisation boîte noire, moindres carrés*

1 Artelys Knitro

Artelys Knitro [2] est le solveur de référence pour les problèmes d'optimisation non-linéaire. Utilisé par des centaines d'industriels et académiques dans des domaines variés (énergie, économie, finance, simulation mécanique, géométrie, etc.) et disponible à travers plus d'une douzaine d'interfaces (AMPL, Matlab, R, C, Python, Java, etc.), Knitro a démontré son efficacité et sa polyvalence dans la résolution de problèmes industriels [1]. Ses 4 algorithmes et ses nombreuses fonctionnalités (différences finies, méthodes de quasi-Newton, tuner) permettent à ses utilisateurs de résoudre efficacement et avec un minimum d'efforts des problèmes d'optimisation non-linéaire de très grande échelle.

Récemment, les challenges d'optimisation en boîte noire BBComp [2] et MINO [3] ont été remportés par des compétiteurs utilisant Knitro, démontrant une fois de plus l'efficacité du solveur.

Les dernières versions ont permis d'introduire de nombreuses fonctionnalités et améliorations de performances. En particulier :

- Une nouvelle spécialisation pour les problèmes de moindres carrés non-linéaires
- Un nouvel algorithme (MISQP [3]) permettant de résoudre des problèmes d'optimisation non-linéaire en variables mixtes (MINLP)
- De nouvelles méthodes de recherche linéaire
- Une nouvelle interface `R` orientée statistiques
- Le scaling utilisateur
- Des performances améliorées pour les 4 algorithmes, les problèmes avec complémentarités (problèmes d'équilibre, MPEC), les méthodes de Quasi-Newton, et l'interface Matlab

La section suivante présente brièvement les travaux réalisés pour les moindres carrés non-linéaires.

2 Moindres carrés non-linéaires

Artelys Knitro 10.2 introduit de nouvelles routines d'optimisation pour les problèmes de moindres carrés non-linéaires, avec contraintes de bornes. Ces problèmes sont très courants en statistiques, data-mining et machine learning. Ils s'expriment sous la forme :

$$\min_p \quad \frac{1}{2}||F(p)||_2^2 \tag{1}$$
$$\text{s.t.} \quad p_L \leq p \leq p_U \tag{2}$$

où p est le vecteur des paramètres (variables à optimiser) et F est une fonction différentiable définissant les résidus.

1. Voir *e.g.*, le benchmark indépendant réalisé par H. Mittelmann : http ://plato.asu.edu/ftp/ampl-nlp.html
2. Black Box Optimization Competition, expensive single-objective track : http ://bbcomp.ini.rub.de/
3. Voir http ://www.mino-itn.unibo.it/challenge-2016/results

L'implémentation réalisée dans Knitro permet de résoudre ces problèmes en utilisant l'un des 4 algorithmes existants dans le solveur (point intérieur/direct, point intérieur/gradient conjugué, active set ou SQP). L'algorithme résultant est une méthode de quasi-Newton dans laquelle, la Hessienne est approximée soit en utilisant l'approximation de Gauss-Newton, soit l'une des autres approximations disponibles dans Knitro (BFGS, L-BFGS, SR1).

L'approximation de Gauss-Newton fournit, à chaque itération, une approximation semi-définie positive de la Hessienne $H(p) \approx J(p)^{\mathsf{T}} J(p)$ (où $J(p)$ est la Jacobienne de la matrice des résidus $F(p)$). Cette approximation a de bonnes propriétés de convergence locale en pratique.

La table 1 présente les résultats de l'algorithme sur un benchmark NIST[6] (gradients calculés par différences finies centrales), ainsi que 2 instances de non-negative least squares issus d'un problème industriel (gradients exacts fournis). Ces résultats sont comparés avec Ceres[1] (version SuiteSparse). Les deux solveurs utilisent le même point initial et leurs paramètres par défaut : approximation de Gauss-Newton et point intérieur direct pour Knitro et Levenberg-Marquardt trust region pour Ceres (et 51 iterations maximum). Les calculs sont réalisés sur un pc portable sous linux équipé d'un core i5-5200U et 8 Gb de mémoire vive.

Difficulty	Problem	n	#res	Artelys Knitro #evalf	#iter	t(ms)	obj	Ceres #evalf	#iter	t(ms)	obj
Lower	Misra1a	2	14	211	28	**1.01**	$6.228 \cdot 10^{-02}$	**103**	**8**	4.7	$6.228 \cdot 10^{-02}$
	Chwirut	3	214	**50**	6	**0.89**	$1.192 \cdot 10^{+03}$	67	**4**	2.2	$1.192 \cdot 10^{+03}$
	Lanczos	6	24	**182**	**12**	**1.01**	$\mathbf{2.704 \cdot 10^{-13}}$	1818	51	4.5	$1.371 \cdot 10^{-09}$
	Gauss	8	250	**119**	6	15.86	$6.579 \cdot 10^{+02}$	140	**4**	**5.0**	$6.579 \cdot 10^{+02}$
	DanWood	2	6	**30**	5	0.25	$2.159 \cdot 10^{-03}$	55	5	0.3	$2.159 \cdot 10^{-03}$
	Misra1b	2	14	**64**	9	159.66	$3.773 \cdot 10^{-02}$	82	**7**	**71.1**	$3.773 \cdot 10^{-02}$
Average	Kirby2	5	151	**111**	9	8.49	$1.953 \cdot 10^{+00}$	138	**6**	**1.0**	$\mathbf{1.953 \cdot 10^{+00}}$
	Hahn	7	236	**151**	**8**	**9.84**	$4.053 \cdot 10^{+00}$	341	11	43.5	$\mathbf{7.662 \cdot 10^{-01}}$
	MGH17	5	33	**120**	9	**0.81**	$5.530 \cdot 10^{-01}$	327	**8**	2.7	$5.530 \cdot 10^{-01}$
	Misra1c	2	14	**31**	**5**	6.69	$2.048 \cdot 10^{-02}$	77	7	**0.7**	$2.048 \cdot 10^{-02}$
	Misra1d	2	14	**29**	**4**	1.72	$2.821 \cdot 10^{-02}$	87	7	**0.5**	$2.821 \cdot 10^{-02}$
	Roszman1	4	25	**45**	4	1.71	$2.474 \cdot 10^{-04}$	76	4	**0.5**	$2.474 \cdot 10^{-04}$
	ENSO	9	168	**400**	20	59.48	$\mathbf{3.943 \cdot 10^{+02}}$	643	**16**	**54**	$3.943 \cdot 10^{+02}$
Higher	Thurber	7	37	**306**	19	10.88	$\mathbf{2.821 \cdot 10^{+03}}$	417	**12**	**4.1**	$2.821 \cdot 10^{+03}$
	BoxBOD	2	6	**83**	14	**0.77**	$\mathbf{5.840 \cdot 10^{+02}}$	230	**10**	2.0	$5.840 \cdot 10^{+02}$
	Rat42	3	9	**81**	**10**	4.77	$\mathbf{4.028 \cdot 10^{+00}}$	186	11	**2.6**	$4.028 \cdot 10^{+00}$
	MGH10	3	16	**83**	8	0.64	$\mathbf{6.834 \cdot 10^{+08}}$	120	8	**0.5**	$1.945 \cdot 10^{+09}$
	Eckerle4	3	35	**51**	**5**	**0.93**	$2.492 \cdot 10^{-01}$	494	25	1.4	$\mathbf{2.491 \cdot 10^{-01}}$
	Rat43	4	15	**133**	**11**	4.81	$1.124 \cdot 10^{+04}$	837	36	4.8	$\mathbf{4.393 \cdot 10^{+03}}$
	Bennett5	3	154	**59**	**7**	**2.64**	$\mathbf{2.620 \cdot 10^{-04}}$	757	51	53.4	$2.640 \cdot 10^{-04}$
Hard	NNLS1	500	200	**13**	**12**	**574.32**	$\mathbf{9.440 \cdot 10^{+03}}$		52	24581.2	$1.438 \cdot 10^{+04}$
	NNLS2	500	1000	**9**	**8**	**819.71**	$\mathbf{2.776 \cdot 10^{+05}}$		48	65176.0	$2.841 \cdot 10^{+05}$

TAB. 1 – Meilleur résultat en gras. n est le nombre de paramètres, *#res* est le nombre de résidus, *#evalf* est le nombre d'évaluations de fonctions (incluant les différences finies), *#iter* est le nombre d'itérations de l'algorithme, *t(ms)* est le temps total (en ms) et *obj* est la valeur finale de l'objectif. Difficulté des instances NIST, voir http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml.

Références

[1] S. Agarwal, K. Mierle, and Others. *Ceres Solver*. http://ceres-solver.org.

[2] R. H. Byrd, J. Nocedal, and R. A. Waltz. *KNITRO : An integrated package for nonlinear optimization.* In G. di Pillo and M. Roma, editors, Large-Scale Nonlinear Optimization, pages 35–59. Springer, 2006.

[3] O. Exler, and K. Schittkowski. *A trust region SQP algorithm for mixed-integer nonlinear programming.* Optimization Letters, Vol. 1, pages 269–280, 2007.

[4] K. Levenberg. *A method for the solution of certain nonlinear problems in least squares*, Quart. Appl. Math, 2(2) :164–168, 1944.

[5] D.W. Marquardt. *An algorithm for least squares estimation of nonlinear parameters*, J. SIAM, 11(2) :431–441, 1963.

[6] F. Mondragon, and B. Borchers. *A Comparison of Nonlinear Regression Codes*, Journal of Modern Applied Statistical Methods, 4(1) :343-351, 2005.