An Algorithm for Constrained Nonlinear Nonsmooth Optimization

Bernard K-S Cheung

GERAD and École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Qc), Canada, H3C 3A7
E-mail: cheung@crt.umontreal.ca

April, 1999

Les Cahiers du GERAD

G–99–25

Abstract

Coupling with the Modified-Grid Unconstrained Optimization Algorithm we introduced in our previous paper (Cheung and Ng, 1995), we have been able to developed a new Algorithm for Nonlinear Optimization capable of working with nonsmooth problems. This Algorithm is based on the Principle of Support by smooth Hypersurfaces. This old but very sound concept has been revitalized by the introduction of a specially constructed Generalized Lagrangian Function and a new penalty updating formula. We have proved without differentiable assumptions that the solution always exists under some mild condition and that the algorithm converges. The Algorithm has performed very well on test runs and on comparison with some popular packages. It offers an easy extension to a global optimization algorithm.

Résumé

Nous avons développé un nouvel algorithme pour l'optimisation non-linéaire de problèmes non-différentiables, basé sur l'algorithme d'optimisation des problèmes sans contraintes introduit dans notre article précédent (Cheung and Ng, 1995). Il a été établi sur le principe de support homogène par hyper-surface. Ce vieux concept, encore très valable, a été revitalisé par une fonction généralisée de Lagrange avec une nouvelle formule de mise à jour de pénalités. Nous avons vérifié que la solution existe, avec une hypothèse de non-différentiabilité et sous de faibles conditions, et que l'algorithme converge. En comparaison avec plusieurs logiciels populaires, cet algorithme performe très bien. Il peut être aisément étendu à l'optimisation globale.

1 Introduction

Since the early 80's, there have been extensive study on nonlinear optimization methods, efficient algorithm such as the Benders Decomposition methods and others have been developed to solve even the very large scale convex problem. For the non-convex especially those non-differentiable problems, some very efficient methods are still lacking. The multipliers method based on the unconstrained minimization of the generalized Lagrangian, (see Nakayama *et al.*, or Mangassanian, 1975) or the augmented Lagrangian (see Bertsekas 1982) developed in the late 70's is still proved to be useful for non-convex constrained optimization problems. However, there seems to be problem of constructing an appropriate Lagrangian and an effective procedure for the updating of both the multiplier and the penalty parameters. As a consequence, this method has been over-shadowed by other powerful methods.

Under exhaustive investigation, we have come up with a truly efficient algorithm using a newly constructed generalized Lagrangian and some special updating up-dating formulae for both the penalty and the multipliers parameters. This algorithm has been shown to be capable of solving some complex non-convex and even non differentiable problems and has compared favorably with some exiting algorithms.

We have test-run our algorithm with some well known test problems, the results are better than or equal to those results obtained by using some popular packages such as IMSL. Our preliminary version is written in C and run on a P.C. We have shown that its performance could be further enhanced by modifying the structure of the generalized Lagrangian function, by using a special Piecewise Concave Updating formula for the penalty-parameter t which provides a close to optimal updating procedure and by a special treatment on its termination criteria. Furthermore, a new *differential weighting* penalty method on problems with different types of constraints is derived and demonstrated to give significant improvement in performance.

The final version has been implemented in a workstation and has been demonstrated to be capable of giving very accurate results which surpass the best recorded in reference literatures on some of the difficult test problems especially those non-convex ones. Although this algorithm in its original form is meant for local search. It can in fact be easily extended into a global optimization method using some additional devices. When coupling with some efficient meta-heuristics, it has demonstrated very good ability in findings the global minimum of a difficult Mixed Integer Nonlinear Programming Problem considered by other well known researcher in this area.

2 Methodology

Consider the following nonlinear programming problem:

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) \\ \text{Subject to} & g_i(\boldsymbol{x}) \geq 0 \qquad i = 1, 2, \ldots, m. \end{array}$$

We can transform it to a sequential Unconstrained Minimization of the following Generalized Lagrangian Function:

$$L(\boldsymbol{x},\boldsymbol{u},t) = f(x) - G(g(\boldsymbol{x}),\boldsymbol{u},t)$$

where $\boldsymbol{u}$ is an m-vector and $g(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}),\ldots,g_m(\boldsymbol{x}))$ and $G(\boldsymbol{v},\boldsymbol{u},t)$ is the function satisfies the following:

(i) $G(\mathbf{0},\boldsymbol{u},t) = 0$ for all $\boldsymbol{u} \geq \mathbf{0}$, $t \geq 0$ and $G(\boldsymbol{v},\mathbf{0},t) = 0$ for all $\boldsymbol{v} \geq \mathbf{0}$, $t \geq 0$.
(ii) $G(\boldsymbol{v},\boldsymbol{u},t)$ is concave and monotonically increasing w.r.t. $\boldsymbol{v}$ for all $\boldsymbol{u} \geq \mathbf{0}$ and $t \geq 0$.
(iii) $G(\boldsymbol{v},\boldsymbol{u},t)$ is convex and monotonically increasing w.r.t. $\boldsymbol{u}$ for all $\boldsymbol{v}$ and $t \geq 0$.
(vi) $G(\mathbf{0},\boldsymbol{u},t)$ tends to a finite limit as v_i tends to $+\infty$ for all $i = 1,2,\ldots,m$.
(v) If $v_i < 0$ for some i, then $G(\mathbf{0},\boldsymbol{u},t)$ tends to $-\infty$ as t tends to ∞.
(vi) If $\boldsymbol{v} > \mathbf{0}$ and $v_i > 0$ for some i, then $G(\mathbf{0},\boldsymbol{u},t)$ tends to 0 as t tends to ∞.
(vii) $\boldsymbol{u} \in \partial_v G(\boldsymbol{v},\boldsymbol{u},t)$ where $\partial_v G(\boldsymbol{v},\boldsymbol{u},t)$ denotes the generalized gradient of G w.r.t. $\boldsymbol{v}$.

Geometrically, this corresponds to the support of the epi-graph of $\beta(y)$ where $\beta(y) = \text{Inf}\{f(\boldsymbol{x}) : \boldsymbol{g}(\boldsymbol{x}) \geq \mathbf{0}\})$ by hypersurfaces. We have proved that if f is Lipschitz and some mild condition on the constraints is satisfied, then the optimal solution exists (see Appendix I). We have succeeded in constructing a convergent Algorithm based on the above principle with smooth supporting hyper-surfaces. A new scheme for updating both the multipliers $\boldsymbol{u}$ and the penalty parameter t, and the Modified Grid Search Method (see Cheung and Ng, 1995) for the unconstrained minimization of the Generalized Lagrangian are employed so that the above constrained problem can be solved readily.

The Modified Grid Search method is based on the original grid search method, but the order of complexity is reduced to n^2, where n is the number of variables. Its local search positions consist of the coordinate points $\boldsymbol{x} \pm h\boldsymbol{e}_i$, where h is the current step size, plus the $2n(n-1)$ positions defined by $\boldsymbol{x} \pm h\boldsymbol{e}_i \pm h\boldsymbol{e}_j$ which lie uniformly on the hypersphere of radius $\sqrt{2}h$ about $\boldsymbol{x}$. It has been shown that if the objective value at $\boldsymbol{x}$ is the lowest comparing with those at the search point, the probability for the actual optimal point to lie within this hypersphere is very high. The other crucial steps consists of a partial local search which helps in refining the search direction and an exploratory movement which allows the search to move along a path (a straight line or a curve, depending on situation) leading it quickly to a position close to the actual optimum point. From the above description and the fact that no differentiability assumption is required for our Modified Grid Algorithm, this constrained algorithm so constructed can be applied to all non-differentiable functions as well, so long as the objective function is Lipschitz continuous.

For problems with some simple constraints, we can remove all those upper bound and lower bound type constraints by simply restricting the the regions of search. It can be proved that such an arrangement will not affect the convergence, and on the contrary, the overall efficiency is improved by reduction of constraints. This assertion has been verified with actual test runs on some examples where in some cases the running time has been reduced by more than 30%.

The detail of the construction is as follows.

1. Construction of $G(\boldsymbol{v}, \boldsymbol{u}, t)$

A convenient form of $G(\boldsymbol{v}, \boldsymbol{u}, t)$ is given as follows:

$$\begin{aligned} G(\boldsymbol{v}, \boldsymbol{u}, t) &= \sum_{i=1}^{m} \varphi(v_i, u_i, t) \\ \text{where } \varphi(v, u, t) &= \begin{cases} uv - (-1)^n tv^n & \text{if } v \le 0 \\ u^2 v/(u + \gamma tv) & \text{if } v > 0 \end{cases} \end{aligned}$$

n being a positive integer ≥ 2, and $\gamma \ge 1$.

Observe that $\varphi(v, u, t)$ is differentiable with respect to v at $v = 0$, this is independent of the choices of n and γ. Hence it is possible to choose the value of n and γ so as to reshape the surface for the best performance. We have found that good performance is obtained with $n = 2$ and $\gamma = 4$.

The procedure involves the unconstrained minimization of the Generalized Lagrangian Function $L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) - G(\boldsymbol{v}, \boldsymbol{u}, t)$ at each augmentation of the value of penalty parameter t followed by updating of the multiplier $\boldsymbol{u}$. The procedure for updating $\boldsymbol{u}$ follows naturally from condition (vii) above. Suppose at the k^{th} iteration the value of $\boldsymbol{u}$ is $\boldsymbol{u}^{(k)}$, then the value in the next iteration is given by $\boldsymbol{u}^{(k+1)} = \partial_v G(\boldsymbol{g}(\boldsymbol{x}^{(k)}), \boldsymbol{u}^{(k)}, t)$, i.e. in our case $u_i^{(k+1)} = u_i^{(k)} - 2\boldsymbol{g}(\boldsymbol{x}^{(k)})t$ if $\boldsymbol{g}(\boldsymbol{x}^{(k)}) \le 0$, and $u_i^{(k+1)} = [u_i^{(k)}/(u_i^{(k)} + 4\boldsymbol{g}(\boldsymbol{x}^{(k)})t)]^2$ for all $i = 1, 2, \ldots, m$. However, it often happens that $\boldsymbol{u}$ is over corrected so that the point of contact is pushed far into the interior of the feasible region. This is detected by $\boldsymbol{g}(\boldsymbol{x}^{(k+1)}) > 0$. One can readjust $u_i^{(k+1)} = (u_i^{(k+1)} + u_i^{(k+2)})/2$ repeatedly so that the point of contact will pulled back to a better position.

2. A new Updating Procedure for the Penalty parameter t

A near optimal produce of updating t has been derived which follows a (monotonically increasing) Piecewise Concave curve as defined in the following.

$$\begin{array}{lllllll} t_1 = 9.125t_0 & t_2 = 14t_0 & t_3 = 17.375t_0 & t_4 = 20.0t_0 \\ t_5 = 110t_0 & t_6 = 164t_0 & t_7 = 200t_0 & \\ t_8 = 850t_0 & t_9 = 1250t_0 & t_{10} = 1500t_0 & \text{and} \end{array}$$

$t_{k+1} = t_k + 80t_0$ for all positive integer $k \ge 10$.

Observe that the sections of the curve from step 0 to step 4, from step 4 to 7 and those from step 7 to step 10 are all logarithmic and becomes linear after step 10. This procedure allows a substantial increase in value of t at certain steps which effectively speed up the search toward the feasible region, while the increase in the subsequence steps becomes more gradual to allow for finer readjustment of the position of the optimal point. Notice also that the value of the penalty parameter t has been increased by 1500 times in ten steps. On actual comparison with the two conventional methods viz (i) $t_{n+1} = \alpha t_n$, with $\alpha > 1$ and (ii) the linear incremental update, we find that this new procedure is much more efficient and works well with a wide variety of problems. As the former method has the disadvantage of having either low increments at the beginning or excessively large increment at the final steps. While the latter method suffers the inflexibility of having only a uniform increment throughout the updating process. The magnitude of the increment should best be readjusted for individual problem. Recently, an additional concave section is added to improve performance on problems required more iterations.

3. Choice of Initial Penalty Parameter t_0

The initial value of the penalty parameter is crucial but not very critical. The optimal value of t_0 depends on the nature of the problem (e.g convex, concave etc.) and the relative magnitude of the objective function to that of the constraints.

As we have seen from the structure of $G(\boldsymbol{v}, \boldsymbol{u}, t)$, the magnitude of t is related to the curvature of the supporting hypersurface, the greater the value of t, the greater is the curvature of the supporting surface in our search region. Hence convex problems require smaller value, while concave problems requires larger value of t_0.

For most of our test examples, especially those of non-convex ones, and where the magnitudes of the objective function and constraints do not differ much, we find the value of $t_0 = 50$ gives very satisfactory results with our new t-updating formula. As for problem # 113, which is convex, we have to use a lower value of t_0, say, 20 or 32 for correct termination. Also, in problem # 64, at the starting point $(1, 1, 1)$, the ratio $f(x)/g(x)$ (where $g(x) \geq 0$ is the active constraint) is larger than 1700, we need an unusually large t0 between 5000 to 10000 for fast and accurate convergence.

It is always advisable to start with a lower value of t_0 , for if its value is too high, it will eventually become very large making the $G(\boldsymbol{v}, \boldsymbol{u}, t)$ term dominating which tends to mask the subtle variation of the objective function, causing it to terminate incorrectly.

One might think that if a problem consists several types of constraints, we should use different values of t_0 for different types of the constraints. This leads to application of *differential weighting procedure* on penalty parameter. The print-out of the part of solution (see Section 2 of results) on a NLP subproblem of a MINLP problems illustrates that a significant improvement has been obtained if the problematic non-convex constraint is given 5 times the initial value of t_0.

4. Choice of Initial Step-size

As reported in our previous study (Cheung and Ng, 1995), initial step-size affect the overall performance of the search algorithm in the following way:

(i) Speed of convergence- normally, larger initial stepsize moves faster, but, if the search path is treacherous or zigzagging, too large a stepsize may have an adverse effect.

(ii) Accuracy of convergence-smaller stepsize gives more accurate results, but slows down the speed of convergence especially when the starting point is far away from the optimal point. Sometimes, too small an initial stepsize may cause instability.

We find an initial size of 0.25 works satisfactorily with most of our test examples. 0.1 also works very well, especially when the starting point is close to the optimal position or when the function fluctuates very much within a small range. We do find a few problems which require an unusually small (say 0.02, or 0.005) or large (say, 2 or 4) step-sizes. There seems to exist an optimal step-size for every problem depending on how the function fluctuates. One could perhaps derive a method which will change the step-size as the condition of the problem requires.

3 Extension to a Global Optimization Algorithm and Other Enhancement

There are several ways of extending our algorithm to a global algorithm.

1. A natural way of doing this is to generate starting point randomly within the feasible region, once the progression is terminated, one can restart with another randomly generated starting point. This procedure is time consuming. However, it turns to be rather successful in some important cases. For example, when it used to evaluate a nonlinear sub-problem of a complex Mixed Integer Nonlinear Programming Problem described in last section of the result. One might easily start the algorithm in parallel with several randomly generated starting points on a modern workstation with multiple processors.
2. Another fairly efficient method is a simple procedure for escaping from the local optimum, this is to restart our search at a point x' directly opposite to the path direction detected before it converges to the suspected local optimal point x^* and at an appropriate distance from it. This distance and the initial step size should be proportional to the length of the directed path. This is described with several test examples in our previous work on nonlinear optimization algorithm with almost simple constraints (see Cheung & Ng 1997). With this arrangement, the probability for our restarted search to reach for the other optimal point is very high, since it can easily reach to the other parts of the region previously untouched by the search. This process can be repeated by generating points at the directions opposite and linearly independent (or orthogonal) to the previous directions of approaches.
3. A more sophisticated way of approaching this problem is to impose some procedures to ensure that those randomly generated starting point are widely and uniformly spreading out over the entire search space. Some of their candidacy potentials can be significantly enhanced by some meta-heuristic search methods (such as the some well designed genetic crossover operations) for a brief period of time. More precisely, the traditional one-point or two-point crossover or some new more sophisticated one helps to produce better fitted starting solutions while causes the initial test points to spread out more thoroughly over the entire search space. The finally selected few of those distinct candidates are to be served as starting points for us to applied our proposed algorithm repeatedly or in parallel for efficiency.

The structure of the Modified Grid Search Algorithm used in the unconstrained minimization of the generalized Lagrangian lends itself to an easy coding for parallel computation using multi-threaded programming in a modern workstation with multiple processors. For example, in the evaluation of $f(\boldsymbol{x} \pm h\boldsymbol{e}_i \pm h\boldsymbol{e}_j)$ for all i, j, one may divide these evaluations into 4 equal but separate groups viz: $f(\boldsymbol{x} + h\boldsymbol{e}_i + h\boldsymbol{e}_j)$ for all $i > j$, $f(\boldsymbol{x} + h\boldsymbol{e}_i - h\boldsymbol{e}_j)$

for all $i > j$, $f(\boldsymbol{x} - h\boldsymbol{e}_i + h\boldsymbol{e}_j)$ for all $i > j$ and $f(\boldsymbol{x} - h\boldsymbol{e}_i - h\boldsymbol{e}_j)$ for all $i > j$. They can now be computed in parallel. There are still other part of the algorithm where computation can be performed in parallel. Thus the efficiency can be enhanced significantly.

4 Results

The first set of results are the test runs on seven selected problems from Test Examples for Nonlinear Programming Code by W. Hock and K. Schittkowski, 1981 and one real life example using our algorithm implemented on a P.C., the IMSL package installed on a VAX and another commercial product for P.C. The results are listed on Table 1 for comparison and followed by some additional remarkable results produced by our final version run on SunSparc Ultra2.

The second set of the results in Table 2 shows the marked improvement after using *differential penalty weighting* on a nonlinear subproblem with a problematic non-convex constraint. Also, the solution of a simple non-differentiable nonlinear problem is solved by our algorithm.

The third set of the result shows that a difficult MINLP problem which arises from chemical engineering and are considered by Floudas, 1989, Kocis and Crossman, 1988 and Salcedo, 1992 is solved by coupling a simple genetic search with our constrained algorithm in 11 minutes on SunSparc Ultra2. Here, the global optimal solution is obtained with no constraint violation[1].

5 Conclusion

Based on the principle of supporting hyper-surfaces, we have succeeded in constructing an efficient algorithm for nonlinear constrained optimization using our specially constructed generalized Lagrangian function and an exotic updating formula for the penalty parameters. When coupling with our modified grid search algorithm, the resulting algorithm is capable of solving some complex non-convex, non-differentiable test problems. The differential penalty weighting method has proved its very good ability in tackling problems with problematic constraints. Early findings indicated that with some simple procedures, it is capable of finding the global optimal solution for problems several local minimums. Further research in this direction will certainly convert this algorithm into a truly global optimum finder.

[1]Note. The list of test problems are to be found in Appendix II.

Table 1: Summary of Comparison Results on Test Functions

Prob. Ref.	**Initial Pt.**		**Eureka Solver**	**IMSL-NCM**	**Modified Grid -Multiplier**
# 38	$(-3, -1, -3, -1)^T$	$x(1)$	1.0000033	1.000004	1.0000000
		$x(2)$	1.0000075	1.0000079	1.0000000
		$x(3)$	0.9999922	0.9999949	1.0000000
		$x(4)$	0.9999434	0.9999903	1.0000000
		$f(x)$	8.828E-10	1.417E-10	2.529E-16
		R^*	0	0	0
		Fn. Eval.	—	462	8527
		Time (sec.)	2	15.488	0.21
# 43	$(0, 0, 0, 0)^T$	$x(1)$	-2.192E-6	-6.091E-8	2.7176E-7
		$x(2)$	1.0000693	1.0000002	0.9999995
		$x(3)$	1.9999820	1.9999999	2.0000000
		$x(4)$	-1.0000076	-0.9999999	-1.0000000
		$f(x)$	-44.000000	-44.000000	-44.000000
		R^*	3.4643E-7	2.8067E-10	0
		Fn. Eval.	—	63	14447
		Time (sec.)	2	12.758	0.12
# 64	$(1, 1, 1)^T$	$x(1)$	108.6718	108.7347	108.7347
		$x(2)$	85.13116	85.12620	85.12621
		$x(3)$	204.3244	204.3246	204.3246
		$f(x)$	6299.8426	6299.84242	6299.84243
		R^*	0	2.800E-13	0
		Fn. Eval.	—	126	6782
		Time (sec.)	10	12.758	0.05
# 104 Optimal Reactor Design	$(6, 3, .4, .2, 6, 6, 1, .5)^T$	$x(1)$	6.4779093	6.4651044	6.465114
		$x(2)$	2.2237887	2.2327319	2.232708
		$x(3)$	0.6670248	0.6673985	0.6673975
		$x(4)$	0.5957162	0.5957575	0.5957564
		$x(5)$	5.9310179	5.9326773	5.932676
		$x(6)$	5.5270681	5.5272319	5.527235
		$x(7)$	1.0107942	1.0133245	1.013322
		$x(8)$	0.4004158	0.4006642	0.4006682
		$f(x^*)$	3.9511602	3.95116344	3.95116344
		R^*	1.5643E-6	5.3817E-11	1.930E-11
		Fn. Eval.	—	164	57302
		Time (sec.)	32	11.219	2.01

Prob. Ref.	**Initial Pt.**		**Eureka Solver**	**IMSL-NCM**	**Modified Grid -Multiplier**
# 108	1	$x(1)$	0.9847108	0.3090008	0.8832961
	1	$x(2)$	0.1741971	-0.9510617	0.5046977
	1	$x(3)$	0.3456377	0.9681441	-0.0054322
	1	$x(4)$	0.9398582	-0.2079283	0.9999852
	1	$x(5)$	0.9847229	0.3090083	0.8832965
	1	$x(6)$	0.1741267	-0.9510617	0.5046971
	1	$x(7)$	0.3415028	0.9781441	-0.0054329
	1	$x(8)$	0.9398787	-0.2079283	0.9999852
	1	$x(9)$	0.0000000	6.8154E-39	0.0000000
		$f(x^*)$	-0.8660225	-0.8660254	-0.866025404
		R^*	8.1782E-12	1.545E-10	8.488E-11
		Fn. Eval.	—	616	57302
		Time (sec.)	22	31.9269	0.78
# 113	2	$x(1)$	2.1830539	2.1719959	2.172073
	3	$x(2)$	2.3335803	2.3636839	2.363495
	5	$x(3)$	8.7560289	8.7739254	8.773894
	5	$x(4)$	5.0464931	5.0959822	5.095974
	1	$x(5)$	1.0498996	0.9906538	0.990706
	2	$x(6)$	1.5830388	1.4305713	1.430779
	7	$x(7)$	1.3446069	1.3216434	1.321792
	8	$x(8)$	9.8487106	9.8287252	9.828846
	9	$x(9)$	8.3292589	8.2800907	8.280227
	10	$x(10)$	8.4245176	8.3759267	8.375972
		$f(x^*)$	24.383302	24.3062091	24.3062095
		R^*	0	5.337E-9	6.835E-9
		Fn. Eval.	—	147	140340
		Time (sec.)	90	16.4688	1.77
Allocation of Community Nurses Problem	114	$x(1)$	Exceed limit, no solution can be found	137.50385	137.5776
	65	$x(2)$		64.80498	64.80889
	4	$x(3)$		4.44515	4.447603
	428	$x(4)$		429.07260	429.3014
	13	$x(5)$		13.47088	13.47423
	2	$x(6)$		2.43907	2.437952
	31.1	$x(7)$		30.26887	30.10000
	30.9	$x(8)$		35.55872	36.60000
	6.5	$x(9)$		10.04582	10.37396
	5	$x(10)$		4.87369	4.900000
	8.3	$x(11)$		8.30834	8.370701
	51	$x(12)$		47.96210	49.38726
		$f(x^*)$		85447.215	85447.004
		R^*		6.188E-9	0
		Fn. Eval.		949	75454
		Time (sec.)		38.33594	2.88

Table 1 (cont.): Some remarkable results obtained by using SunSparc Ultra2

Problem # 104

Minimum solution obtained:

$X^*(1)$	=	6.465114
$X^*(2)$	=	2.232709
$X^*(3)$	=	0.6673975
$X^*(4)$	=	0.5957564
$X^*(5)$	=	5.932676
$X^*(6)$	=	5.527235
$X^*(7)$	=	1.013322
$X^*(8)$	=	0.4006683
Minimum function value	=	3.95116344
No. of iterations	=	21
Total no. of function eval.	=	77776
Total no. of min. fct. swap	=	5046
Sum of constraint violations	=	0
$U[1]$	=	2.35981
$U[2]$	=	6.20551
$U[3]$	=	0.92764
$U[4]$	=	0.847196
Active constraints	=	(1,2,3,4,-)
Run time	=	0.5 sec.

Problem # 108

Minimum solution obtained:

$X^*(1)$	=	0.863297
$X^*(2)$	=	0.5046963
$X^*(3)$	=	-0.005431441
$X^*(4)$	=	0.9999852
$X^*(5)$	=	0.8632969
$X^*(6)$	=	0.5046964
$X^*(7)$	=	-0.005431331
$X^*(8)$	=	0.9999853
$X^*(9)$	=	0
Minimum function value	=	-0.866025404
No. of iterations	=	11
Total no. of function eval.	=	56884
Total no. of min. fct. swap	=	3718
Sum of constraint violations	=	-7.58746e-11
$U[1]$	=	0.144338
$U[3]$	=	0.144337
$U[4]$	=	0.144337
$U[6]$	=	0.144337
$U[7]$	=	0.144338
$U[9]$	=	0.144338
$U[11]$	=	0.00012774
$U[13]$	=	0.0142214
Active constraints = (1,3,4,6,7,9.11,13,-)		
Run time	=	0.1 sec.

Problem # 113

Minimum solution obtained:

$X^*(1)$	=	2.172072
$X^*(2)$	=	2.363496
$X^*(3)$	=	8.773894
$X^*(4)$	=	5.095975
$X^*(5)$	=	0.9907055
$X^*(6)$	=	1.430778
$X^*(7)$	=	1.321791
$X^*(8)$	=	9.828845
$X^*(9)$	=	8.280175
$X^*(10)$	=	8.375645
Minimum function value	=	24.3062095
No. of iterations	=	13
Total no. of function eval.	=	123206
Total no. of min. fct. swap	=	7665
Sum of constraint violations	=	0
$U[1]$	=	1.7164
$U[2]$	=	0.474673
$U[3]$	=	1.37604
$U[4]$	=	0.0205562
$U[5]$	=	0.312278
$U[7]$	=	0.28732
Active constraints	=	(1,2,3,4,-)
Run time	=	0.5 sec.

Table 2: Constrained Minimization Linked with Grid III

Problem # CP11 [Nondifferentiable Problem]

No. of variables	=	2
No. of constraints	=	2
Initial step length	=	0.250000
Contraction Factor	=	4
Initial t	=	50
Tolerance	=	le-07

Initial Point:
$Xo[1]$ = 5
$Xo[2]$ = 0.5
Minimum solution obtained:
$X^*[1]$ = 4
$X^*[2]$ = 0

Minimum function value	=	-1
No. of iterations	=	3
Total no. of function eval.	=	318
Total no. of min. fct. swap	=	40
Final t	=	456

Lagrange multiplier:
$U[1]$ = 0
$U[2]$ = 0
Difference:
$G[1]$ = 2.3094
$G[2]$ = 2

Sum of constraint violations	=	0
Elapsed time (in seconds)	=	0.000000

Problem # CP11 [Nondifferentiable Problem]

No. of variables	=	2
No. of constraints	=	2
Initial step length	=	0.250000
Contraction Factor	=	4
Initial t	=	50
Tolerance	=	le-08

Initial Point:
$Xo[1]$ = 3
$Xo[2]$ = 2
Minimum solution obtained:
$X^*[1]$ = 3
$X^*[2]$ = 1.732051

Minimum function value	=	-0.999999998
No. of iterations	=	6
Total no. of function eval.	=	907
Total no. of min. fct. swap	=	140
Final t	=	869

Lagrange multiplier:
$U[1]$ = 0.866016
$U[2]$ = 0.499986
Difference:
$G[1]$ = 1.28493e-09
$G[2]$ = 2.22556e-09

Sum of constraint violations	=	0
Elapsed time (in seconds)	=	0.054945

Problem # CP15 [MINLP-Problem]

No. of variables	=	7
No. of constraints	=	7
Initial step length	=	0.100000
Contraction Factor	=	5
Initial t	=	32000
Tolerance	=	5e-07

Initial Point:
$Xo[1]$ = 1000
$Xo[2]$ = 1000
$Xo[3]$ = 1000
$Xo[4]$ = 250
$Xo[5]$ = 150
$Xo[6]$ = 20
$Xo[7]$ = 16
Minimum solution obtained:
$X^*[1]$ = 479.9996
$X^*[2]$ = 719.9994
$X^*[3]$ = 959.9949
$X^*[4]$ = 239.9987
$X^*[5]$ = 119.9999
$X^*[6]$ = 20
$X^*[7]$ = 16

Minimum function value	=	38499.40469
No. of iterations	=	9
Total no. of function eval.	=	37490
Total no. of min. fct. swap	=	791
Final t	=	6400000
Sum of constraint violations	=	-5.21386e-05
Time (in seconds)	=	0.29

Result before application of the differential penalty procedure:

Minimum solution obtained:
$X^*[1]$ = 479.5364
$X^*[2]$ = 719.3048
$X^*[3]$ = 959.0723
$X^*[4]$ = 239.768
$X^*[5]$ = 119.8842
$X^*[6]$ = 20
$X^*[7]$ = 16

Minimum function value	=	38477.1481
No. of iterations	=	51
Total no. of function eval.	=	259135
Total no. of min. fct. swap	=	16977
Sum of constraint violations	=	-0.0032601
Time (in seconds)	=	1.73

Table 3: Solution of an MINLP Problem

No. of generations	=	60
Population size	=	5
Reproduction rate	=	0.950000
Mutation (rate)	=	0.020000
Shuffle tolerance	=	60
Best solution:		
$(N_1, N_2, \ldots, N_3)$	=	(2, 2, 3, 2, 1, 1)
V_1	=	2999.999999
V_2	=	1894.312956
V_3	=	1974.683537
V_4	=	2622.781258
V_5	=	2330.195057
V_6	=	2111.739265
T_{L1}	=	3.200001
T_{L2}	=	3.400002
T_{L3}	=	6.200001
T_{L4}	=	3.400004
T_{L5}	=	3.700001
B_1	=	379.746824
B_2	=	771.406254
B_3	=	728.185956
B_4	=	638.297868
B_5	=	526.198045
Best solution value	=	285622.72879
Run time	=	11:19.7

References

[1] Bertsekas, D.P., *Constrained Optimization and Lagrange Multiplier Method*, Academic Press, 1982

[2] Cheung, B.K.S., Ng, A.C.L., “An Efficient and Reliable Algorithm for Nonsmooth Nonlinear Optimization”m *Neural, Parallel & Scientific Computations* 3, 115–128, 1995.

[3] Cheung, B.K.S., Ng, A.C.L., “An Efficient Search Method for Nonsmooth Nonlinear Optimization Problems with Mostly Simple Constraints”, *Neural, Parallel & Scientific Computations* 5, 335–346, 1997.

[4] Cheung, B.K.S., Langevin, A., Delmaire, H., “Coupling Genetic Algorithm with a Grid Search Method to Solve Mixed Integer Nonlinear Programming Problems”, *Computers & Mathematics with Applications* 34(12), 13–23, 1997.

[5] Clark, F.H., *Optimization and Nonsmooth Analysis*, Wiley, 1983.

[6] Geoffrion, A.M., “Generalized Benders’ Decomposition”, *J. Optim. Theory & Appl.* 10, 237, 1972.

[7] Floudas, C.A., Ciric, A.R., "Strategies for Overcomign Uncertainty in Heat Exchanger Network Synthesis", *Comput. Chem. Eng. J.* 13(10), 1133–1152, 1989.

[8] Floudas, C.A., Pardalos, P.M. *A Collection of Test Problems for Constrained Global Optimization Algorithms*, Springer-Verlag, 1987.

[9] Hock, W., Schittkowski, K., *Test Examples for Nonlinear Programming Code*, Springer-Verlag, 1981.

[10] Mangassanian, O.L., "Unconstrained Lagrangian in Nonlinear Programming", *SIAM Journal of Control* 13, 772–791, 1975.

[11] Nakayama, H., Sayama, H., Sawargi, Y., "A Generalized Lagrangian Function and Multipliers Method", *J.O.T.A.* 17(3/4), 221–227, 1975.

[12] Kocis, G.R., Grossman, I.E., "Global Optimization of Non-convex Mixed Integer Nonlinear Programming Problems in Process Synthesis", *Ind. Eng. Chem. Res.* 27, 1407–1421, 1988.

[13] Salcedo, R.L., "Solving Non-convex Nonlinear Programming and Mixed-integer Nonlinear Programming Problems with Adaptive Random Search", *Ing. Eng. Chem. Res.* 31, 2622-273, 1992.

Appendix I

On Principle of Supporting Hyper-surface and Algorithm Convergence

Consider the following problem

$$(P)\quad \begin{array}{lll} \text{Minimized} & f(\boldsymbol{x}) & \\ \text{Subject to} & g_i(\boldsymbol{x}) \geq 0 & i = 1, 2, 3, \ldots, m \end{array}$$

where f and g_i are Lipschitz near $\boldsymbol{x}$, for all $\boldsymbol{x}$ lying on an open subset $\cup$ of $\Re^n$ containing the feasible region defined by $g_i(\boldsymbol{x}) \geq 0, i = 1, 2, \ldots, m$.

The perturbed problem (P_y) of (P) is defined as follows.

$$(P_y)\quad \begin{array}{lll} \text{Minimized} & f(\boldsymbol{x}) & \\ \text{Subject to} & g_i(\boldsymbol{x}) \geq y_i & i = 1, 2, 3, \ldots, m \end{array}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \Re^{m+}$.

We shall assume that (P_y) is solvable for all y in a neighbourhood of 0. Let $E = \{\boldsymbol{y} : \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{y}$ for some $\boldsymbol{x} \in \Re^n\}$ where $\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x}))$ and defined a function $\beta(\boldsymbol{y}) = \inf\{f(\boldsymbol{x}) : \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{y}\}$. The epigraph of $\beta(\boldsymbol{y})$ is the set epi $\beta(\boldsymbol{y}) = \{(\boldsymbol{y}, \boldsymbol{z}) : \boldsymbol{z} \geq \beta(\boldsymbol{y}), \boldsymbol{y} \in E\}$. It is not difficult to show that the following is true.

Proposition 1

(i) $\beta(\boldsymbol{y})$ is nondecreasing and $\beta(\boldsymbol{y})$ is the constrained minimum of (P_y), if it is solvable.

(ii) If $f(\boldsymbol{x})$ is convex and g_i's are concave for each i, then $\beta(\boldsymbol{y})$ is also convex.

(iii) If $\boldsymbol{x}^*$ solves (P), then $\beta(t\boldsymbol{g}(\boldsymbol{x}^*)) = f(\boldsymbol{x}^*)$ for $0 \leq t \leq 1$.

Let G be a mapping of $\Re^m \times \Re^{m+} \times \Re^+$ into $\Re^n$ satisfying the following properties:

(i) $G(\boldsymbol{0}, \boldsymbol{u}, t) = 0$ for all $\boldsymbol{u} \geq \boldsymbol{0}$, $t \geq 0$ and $G(\boldsymbol{v}, \boldsymbol{0}, t) = 0$ for all $\boldsymbol{v} \geq \boldsymbol{0}$, $t \geq 0$.

(ii) $G(\boldsymbol{v}, \boldsymbol{u}, t)$ is concave and monotonically increasing w.r.t. $\boldsymbol{v}$ for all $\boldsymbol{u} \geq \boldsymbol{0}$ and $t \geq 0$.

(iii) $G(\boldsymbol{v}, \boldsymbol{u}, t)$ is convex and monotonically increasing w.r.t. $\boldsymbol{u}$ for all $\boldsymbol{v}$ and $t \geq 0$.

(iv) $G(\boldsymbol{0}, \boldsymbol{u}, t)$ tends to a finite limit as v_i tends to $+\infty$ for all $i = 1, 2, \ldots, m$.

(v) If $v_i < 0$ for some i, then $G(\boldsymbol{0}, \boldsymbol{u}, t)$ tends to $-\infty$ as t tends to ∞.

(vi) If $\boldsymbol{v} \geq \boldsymbol{0}$ and $v_i > 0$ for some i, then $G(\boldsymbol{0}, \boldsymbol{u}, t)$ tends to 0 as t tends to ∞.

(vii) $\boldsymbol{u} \in \partial_v G(\boldsymbol{v}, \boldsymbol{u}, t)$ where $\partial_v G(\boldsymbol{v}, \boldsymbol{u}, t)$ denotes the generalized gradient of g w.r.t. $\boldsymbol{v}$.

Now, we define Lagrangian function as $L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) - G(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{u}, t)$.

Let $h(\boldsymbol{u}, t) = \inf_{x \in \Re^n} L(\boldsymbol{x}, \boldsymbol{u}, t)$.

The Lagrangian Dual Problem is formulated as the following:

$$(D)\quad \begin{array}{ll} \text{Maximize} & h(\boldsymbol{u}, t) \\ \text{Subject to} & \boldsymbol{u} \in \Re^{m+} \end{array}$$

Geometrically, one may think of $z - G(\boldsymbol{v}, \boldsymbol{u}, t) = k(\boldsymbol{v}', \boldsymbol{z}')$ for fixed $\boldsymbol{u}$ and t as a curved surface passing through the point $(\boldsymbol{v}', \boldsymbol{z}')$ making intercept $k(\boldsymbol{v}', \boldsymbol{z}')$ with the z-axis. One has the following important result.

Proposition 2 *$\boldsymbol{x}$ minimizes $L(\boldsymbol{x}, \boldsymbol{u}, t)$ if and only if the hyper-surface $z - G(\boldsymbol{v}, \boldsymbol{u}, t) = k(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$ supports epi $\beta(\boldsymbol{y})$ at $(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$ in the sense that $z - G(\boldsymbol{v}, \boldsymbol{u}, t) \geq f(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$, for all $(\boldsymbol{y}, z) \in$ epi $\beta(\boldsymbol{y})$.*

Proof: If $z - G(\boldsymbol{v}, \boldsymbol{u}, t) = k(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$ supports epi $\beta(\boldsymbol{y})$ at $(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$, we have $L(\boldsymbol{x}', \boldsymbol{u}, t) = f(\boldsymbol{x}) - G(\boldsymbol{g}(\boldsymbol{x}'), \boldsymbol{u}, t,) = k(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$, and clearly $(\boldsymbol{g}(\boldsymbol{x}), f(\boldsymbol{x})) \in$ epi $\beta(\boldsymbol{y})$, this implies that

$$L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) - G(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{u}, t) \geq f(\boldsymbol{x}') - G(\boldsymbol{g}(\boldsymbol{x}'), \boldsymbol{u}, t) = L(\boldsymbol{x}', \boldsymbol{u}, t).$$

Conversely, if $\boldsymbol{x}'$ minimizes $L(\boldsymbol{x}, \boldsymbol{u}, t)$, then for all $(\boldsymbol{y}, z) \in$ epi $\beta(\boldsymbol{y})$, $z \geq \inf\{f(\boldsymbol{x}), \forall \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{y}\}$. Hence, for all $\epsilon > 0, z + \epsilon > f(\underline{\boldsymbol{x}})$ for some $\underline{\boldsymbol{x}}$ such that $\boldsymbol{g}(\underline{\boldsymbol{x}}) \geq \boldsymbol{y}$, this implies that $z - G(\boldsymbol{v}, \boldsymbol{u}, t) + \epsilon > f(\underline{\boldsymbol{x}}) - G(\boldsymbol{g}(\underline{\boldsymbol{x}}), \boldsymbol{u}, t) \geq f(\boldsymbol{x}') - G(\boldsymbol{g}(\boldsymbol{x}'), \boldsymbol{u}, t)$. Since ϵ is arbitrary, we have $z - G(\boldsymbol{v}, \boldsymbol{u}, t) \geq f(\boldsymbol{x}') - G(\boldsymbol{g}(\boldsymbol{x}'), \boldsymbol{u}, t)$. Thus the hyper-surface supports epi $\beta(\boldsymbol{y})$ at $(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$. □

From the above proposition, one obtained readily the follow equivalences.

(a) $\boldsymbol{x}^*$ is a solution of the problem (P) and the hyper-surface $z - G(\boldsymbol{v}, \boldsymbol{u}^*, t) = k(\boldsymbol{g}(\boldsymbol{x}'), f(\boldsymbol{x}'))$ supports epi $\beta(\boldsymbol{y})$ at $(0, f(\boldsymbol{x}^*))$.
(b) $\boldsymbol{x}^*$ minimizes $L(\boldsymbol{x}, \boldsymbol{u}^*, t)$, for sufficiently large t and $g(\boldsymbol{x}^*) \geq 0$ and $G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t) = 0$.
(c) $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is a saddle point solution of $L(\boldsymbol{x}, \boldsymbol{u}, t)$.
(d) $\boldsymbol{x}^*$ is a minimum of the problem (P), while $\boldsymbol{u}^*$ is the maximum of its dual problem (D), and $f(\boldsymbol{x}^*) = h_t(\boldsymbol{u}^*)$.

Definition. Let $\boldsymbol{x}^*$ solves (P), the problem (P) is said to be calm at $\boldsymbol{x}^*$, provided there exists positive real numbers ϵ and M such that for all $\boldsymbol{y}$ with $\|\boldsymbol{y}\| \geq \epsilon$ and for all $\boldsymbol{x}$ such that (P_y) is feasible and $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \epsilon$,

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) + M\|\boldsymbol{y}\| \geq 0. \tag{1}$$

Note. The note of calmness was introduced by F. Clark (see Clark 1983). Not every problem is calm. E.g. the following problem:

$$\begin{array}{lrcl} \text{Minimize} & f(x_1, x_2) & = & -x_1 \\ \text{Subject to} & g_1(x_1, x_2) & = & x_1 \geq 0 \\ & g_2(x_1, x_2) & = & x_2 \geq 0 \\ & g_3(x_1, x_2) & = & (1 - x_1)^3 - x_2 \geq 0 \end{array}$$

has an obvious minimum $\boldsymbol{x}^* = (1, 0)$, but is not calm at $(1, 0)$. For if we take $\boldsymbol{y} = (0, 0, y_3)$ with $y_3 \neq 0$, from the inequality constraint g_3, we see that $\beta(\boldsymbol{y}) - \beta(\mathbf{0}) = y_3^{1/3}$. We have $\frac{\partial \beta(\boldsymbol{y})}{\partial y_3} = \frac{1}{3y_3^{3/2}}$ which tends to ∞ as y_3 tends to 0. the relation (1) is clearly not satisfied.

Theorem 1 *If $\boldsymbol{x}^*$ is a solution of problem (P), and (P) is calm at $\boldsymbol{x}^*$, then there exists a generalized Lagrangian function $L(\boldsymbol{x}, \boldsymbol{u}, t)$ satisfying assumption (i)-(vi) such that $\boldsymbol{x}^*$ minimizes $L(\boldsymbol{x}, \boldsymbol{u}, t)$, for sufficiently large t and $L(\boldsymbol{x}^*, \boldsymbol{u}, t) = f(\boldsymbol{x}^*)$.*

Proof. Since (P) is calm at $\boldsymbol{x}^*$, there exist real numbers ϵ and M such that, for all $\boldsymbol{y}, \|\boldsymbol{y}\| \leq \epsilon$ and for all $\boldsymbol{x}$ where (P_y) is feasible and $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \epsilon$

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) + M\|\boldsymbol{y}\| \geq 0$$

Let $L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) + t\left[\sum_{i=1}^{m} \min\{0, g_i(\boldsymbol{x})\}\right]$.

Now, for all $\boldsymbol{x}$, such that $\boldsymbol{g}(\boldsymbol{x}) \not\geq \mathbf{0}$, $\|\boldsymbol{g}(\boldsymbol{x})\| < \epsilon$ and $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \epsilon$, $L(\boldsymbol{x}, \boldsymbol{u}, t) - L(\boldsymbol{x}^*, \boldsymbol{u}, t) = f(\boldsymbol{x}) - f(\boldsymbol{x}^*) - t\left[\sum_{i=1}^{m} \min\{0, g_i(\boldsymbol{x})\}\right] \geq f(\boldsymbol{x}) - f(\boldsymbol{x}^*) - t\|\boldsymbol{y}\|$, where $\boldsymbol{y}$ is the vector such that $y_i = g_i(\boldsymbol{x})$, if $g_i(\boldsymbol{x}) < 0$ and $y_i = 0$, if $g_i(\boldsymbol{x}) \geq 0$. Obviously, $\|\boldsymbol{y}\| \leq -\left[\sum_{i=1}^{m} \min\{0, \boldsymbol{g}_i(\boldsymbol{x})\}\right] \leq \|\boldsymbol{g}(\boldsymbol{x})\|$. Choose $t > M$, we have $L(\boldsymbol{x}, \boldsymbol{u}, t) - L(\boldsymbol{x}^*, \boldsymbol{u}, t) \geq f(\boldsymbol{x}) - f(\boldsymbol{x}^*) - M\|\boldsymbol{y}\| \geq 0$.

It is clear that for all $\boldsymbol{x}$ such that $g(\boldsymbol{x}) \geq \mathbf{0}, L(\boldsymbol{x}, \boldsymbol{u}, t) - L(\boldsymbol{x}^*, \boldsymbol{u}, t) \geq 0$. Moreover, $L(\boldsymbol{x}^*, \boldsymbol{u}, t) = f(\boldsymbol{x}^*)$, since $g(\boldsymbol{x}^*) \geq \mathbf{0}$. □

Theorem 2 *(Generalized Kuhn Tuck Condition). If $\boldsymbol{x}$ is a solution of problem (P), and (P) is calm, then there exists an m-vector $\boldsymbol{u}^* \geq \mathbf{0}$ such that $\mathbf{0} \in \partial f(\boldsymbol{x}) - \boldsymbol{u}^*\partial \boldsymbol{g}(\boldsymbol{x})$, where $\partial f(\boldsymbol{x})$ is the subdifferential of f at $\boldsymbol{x}$, and $\boldsymbol{u}^* \cdot \boldsymbol{g}(\boldsymbol{x}^*) = \mathbf{0}$ (i.e. $\exists \eta \in \partial f(\boldsymbol{x}^*)$ and $\xi_i \in \partial g_i(\boldsymbol{x}^*)$ such that $\eta = \Sigma u_i^* \xi_i$ and $u_i^* g_i(\boldsymbol{x}^*) = 0$ for $i = 1, 2, \ldots, m$.).*

Proof. By Theorem 2, there is a generalized Lagrangian function $L(\boldsymbol{x}, \boldsymbol{u}, t)$ such that $\boldsymbol{x}^*$ minimized $L(\boldsymbol{x}, \boldsymbol{u}, t)$ for sufficiently large t and $L(\boldsymbol{x}^*, \boldsymbol{u}, t) = f(\boldsymbol{x}^*)$. Since this function is Lipschitz in the neighbourhood of $\boldsymbol{v} = \boldsymbol{g}(\boldsymbol{x}^*)$, we have $\mathbf{0} \in \partial_x L(\boldsymbol{x}^*, \boldsymbol{u}, t) - \partial f(\boldsymbol{x}^*) - \partial_v G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$. Hence $\mathbf{0} = \eta - \lambda$ for some $\eta \in \partial f(\boldsymbol{x}^*)$ and $\partial \in \partial_x G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$. By the chain rule of sub-differentials, $\lambda = \Sigma u_i^* \xi_i i = 1, 2, \ldots, m$, where $\boldsymbol{u}^*(u_1, u_2, \ldots, u_m) \in \partial_v G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$ and $\xi_i \in \partial g_i(\boldsymbol{x}^*)$. Since G is monotonic increasing w.r.t. $\boldsymbol{u}$, we must have $\boldsymbol{u}^* \geq \mathbf{0}$. It remains to show that $u_i^* g_i(\boldsymbol{x}^*) = 0$ for $i = 1, 2, \ldots, m$. $G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t) = 0$ and G is concave w.r.t. v_i for all i and $g_i(\boldsymbol{x}^*) > 0$ imply that $0 \in \partial_{vi} G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$. (Since $G(\mathbf{0}, \boldsymbol{u}^*, t) = 0$. Here we may identify the i^{th} component of $\partial_v G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$ with $\partial_{v_i} G(\boldsymbol{g}(\boldsymbol{x}^*), \boldsymbol{u}^*, t)$). Thus $u_i^* = 0$, if $g_i(\boldsymbol{x}^*) > 0$. So, we have proved $u_i^* g_i(\boldsymbol{x}^*) = 0$ for $i = 1, 2, \ldots, m$. □

The generalized Kuhn Tucker condition was first proved by F. Clarke (see Clarke 1983), for nonlinear problem in Hilbert space which requires a more lengthy reasoning. Now, we have provided a simple alternative proof in Euclidean space.

Note. Observe that the classical Lagrangian function $L(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) - \boldsymbol{u} \cdot \boldsymbol{g}(\boldsymbol{x})$ is a special case of the above function.

The following theorem is crucial for application of this principle of supporting hypersurfaces to the solution of non-convex constrained optimization problems.

Theorem 3 *Let $L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) - G(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{u}, t)$ be a generalized Lagrangian function such that the function $G(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{u}, t)$ satisfies the additional assumption (VIII) which requires that $\boldsymbol{u} \in \partial_v G(\mathbf{0}, \boldsymbol{u}^*, t)$ where $\partial_v G(\boldsymbol{v}, \boldsymbol{u}^*, t)$ denotes the generalized gradient of G with* *respect to $\boldsymbol{v}$ with $\boldsymbol{u}$ and t fixed. Suppose $\boldsymbol{x}^*$ solves (P) and there is a neighbourhood N_0 of $\mathbf{0}$, such that $\beta(\boldsymbol{y}) - \beta(0) \geq \boldsymbol{u}' \cdot \boldsymbol{y}$ for some $\boldsymbol{u}' \geq \mathbf{0}$ and for all $\boldsymbol{y} \in N_0$, then for every positive real number M, $\boldsymbol{x}^*$ minimizes $L(\boldsymbol{x}, \boldsymbol{u}, t)$ for sufficiently large t and for all $\boldsymbol{x}$ such that $0 \leq g_i(\boldsymbol{x}) \leq M, i = 1, 2, 3, \ldots, m$.* □

The proof of this theorem is described in detail in Nakayama *et al.* in 1975. One would notice that validity of this theorem does not require epi $\beta(\boldsymbol{y})$ to be locally convex. So long as epi $\beta(\boldsymbol{y})$ can be supported locally by a hyperplane, the sequential minimization of the generalized Lagrangian will solve the constrained problem (P).

Remark. Theorem 1 can be proved by a modification of our original method, if the Lagrangian is replaced by the following one.

$L(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x}) - \sum_{i=1}^{m} u_i \min\{0, \boldsymbol{g}_i(\boldsymbol{x})\} - t(\min\{0, \boldsymbol{g}_i(\boldsymbol{x})\})^2$ which is smooth everywhere except at the points $\boldsymbol{v} \geq \mathbf{0}$ with $v_i = 0$ for some i.

Support by Smooth Hyper-surfaces

There is no restriction to the use of smooth hyper-surfaces, since Theorem 3 applies to all types of function $G(\boldsymbol{v}, \boldsymbol{u}, t)$ satisfying all the assumptions. The function G described in the main text is a good example. More generally, if G is twice differentiable, then the assumption (VIII) becomes $\nabla_u G(\mathbf{0}, \boldsymbol{u}, t) = \boldsymbol{u}$, the updating of the multiplier $\boldsymbol{u}$ can easily be determined and on the whole, the process of sequential unconstrained minimization of the Lagrangian of the problem (P) should be smoother, even if the objective function and the constraints are non-differentiable.

We consider the function G which has the form $G(\boldsymbol{v}, \boldsymbol{u}, t) = \sum_{i=0}^{m} \Phi(v_i, u_i, t)$ where $\Phi : \Re^2 \times \Re^+ \rightarrow \Re$ is a piecewise smooth function satisfying the following:

(i) $\Phi(0, u, t) = 0$ for all $u \geq 0$, $t > 0$.
(ii) $\Phi(v, 0, t) = 0$ for all $v \geq 0$, $t > 0$.
(iii) $\Phi(v, u, t)$ is strictly concave and monotonic increasing with respect to v for all $u \geq 0$ and $t > 0$.
(iv) $\Phi(v, u, t)$ is strictly convex with respect to u for all v and $t > 0$.
(v) $\Phi(v, u, t)$ tends to a finite limit as $v \rightarrow +\infty$.
(vi) If $v < 0$, $\Phi(v, u, t) \rightarrow -\infty$, as $t \rightarrow +\infty$.
(vii) If $v < 0$, $\Phi(v, u, t) \rightarrow 0$, as $t \rightarrow -\infty$.
(viii) $\Phi_v^-(o, u, t) = u$ for all $u \geq 0$ and $t > 0$ (where $\Phi_v^-(o, u, t)$ is the left hand derivative of Φ with respect to v and 0 and this coincide with it partial derivative w.r.t. if Φ is differentiable at 0).

Now, we shall show that the sequential minimization of our Lagrangian function with $G(\boldsymbol{v}, \boldsymbol{u}, t)$ defined in the above paragraph with continuous updating of $\boldsymbol{u}$ following by updating of the penalty parameter t produces a monotonic increasing sequence $h(\boldsymbol{u}^{(k)}, t^{(k)})$ which converges even if (P) is nonconvex or non-differentiable.

Let $\boldsymbol{u}^{(k+1)}$ be updated vector value of $\boldsymbol{u}^{(k)}$ at the k^{th} iteration, and $t^{(k+1)}$ be the subsequent updated value of $t^{(k)}$, and let $\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k+1)}$ be the respective minima of the

Lagrangians $L(\boldsymbol{x}, \boldsymbol{u}^{(k)}, t^{(k)})$ and $L(\boldsymbol{x}, \boldsymbol{u}^{(k+1)}, t^{(k)})$. We have,

$$
\begin{aligned}
&h(\boldsymbol{u}^{(k+1)}, t^{(k)}) - h(\boldsymbol{u}^{(k)}, t^{(k)}) \\
&\quad = L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) - L(\boldsymbol{x}^{(k)}, \boldsymbol{u}^{(k)}, t^{(k)}) \\
&\quad = \left[L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) - L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k)}, t^{(k)})\right] \\
&\qquad\quad + \left[L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k)}, t^{(k)}) - L(\boldsymbol{x}^{(k)}, \boldsymbol{u}^{(k)}, t^{(k)})\right] \\
&\quad \geq \left[L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) - L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k)}, t^{(k)})\right] \\
&\qquad\quad (\text{since } \boldsymbol{x}^{(k)} \text{ minimizes } L(\boldsymbol{x}^{(k)}, \boldsymbol{u}^{(k)}, t^{(k)})) \\
&\quad \geq G(g(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k)}, t^{(k)}) - G(g(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) \\
&\quad \geq \nabla_{\boldsymbol{u}}^{-} G(g(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) \cdot (\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k+1)}) \\
&\quad = \sum_{i=1}^{m} \Phi_{i_{\boldsymbol{u}}}^{-}(g_i(\boldsymbol{x}^{(k+1)}), \boldsymbol{u}_i^{(k+1)}, t^{(k)})(u_i^{(k)} - u_i^{(k+1)}) \geq 0.
\end{aligned}
$$

It follows from the fact that $\Phi(v, u, t)$ is convex w.r.t. u and $\Phi(v, 0, t) = 0$, which implies that if $g_i(\boldsymbol{x}^{(k+1)}) \leq 0$, then $\Phi_{i_{\boldsymbol{u}}}^{-}(g_i(\boldsymbol{x}^{(k+1)}), u_i^{(k+1)}, t^{(k)}) \geq 0$ and $u_i^{(k+1)} \leq u_i^{(k)}$.

Now, if $t^{(k)}$ is updated to $t^{(k+1)}$, and $\underline{\boldsymbol{x}}^{(k+1)}$ minimizes $L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k+1)})$, then

$$
\begin{aligned}
&h(\boldsymbol{u}^{(k+1)}, t^{(k+1)}) - h(\boldsymbol{u}^{(k+1)}, t^{(k)}) \\
&\quad = L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k+1)}) - L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) \\
&\quad = \left[L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k+1)}) - L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)})\right] \\
&\qquad\quad + \left[L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) - L(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)})\right] \\
&\quad \geq L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k+1)}) - L(\underline{\boldsymbol{x}}^{(k+1)}, \boldsymbol{u}^{(k+1)}, t^{(k)}) \\
&\qquad\quad \text{as } \boldsymbol{x}^{(k+1)} \text{ minimizes } L(\boldsymbol{x}, \boldsymbol{u}^{(k+1)}, t^{(k)}), \\
&\quad = G(g(\underline{\boldsymbol{x}}^{(k+1)}), \boldsymbol{u}^{(k+1)}, t^{(k)}) - G(g(\underline{\boldsymbol{x}}^{(k+1)}), \boldsymbol{u}^{(k+1)}, t^{(k+1)}) \geq 0 \text{ for sufficiently large } t^{(k)}.
\end{aligned}
$$

This follows from (vii) and that $g(\underline{\boldsymbol{x}}^{(k+1)}) \leq 0$.

Hence the sequence $\{h(\boldsymbol{u}^{(k)}, t^{(k)})\}$ is monotonic increasing, since $h(\boldsymbol{u}^{(k)}, t^{(k)}) \geq h(\boldsymbol{u}^*, t)$ for all k and for sufficiently large t, this sequence converges. By our construction, if $\underline{\boldsymbol{u}} = (\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_m)$ and $\underline{t}$ are such that $h(\underline{\boldsymbol{u}}, \underline{t})$ is the limit of the above sequence and $\underline{\boldsymbol{x}}$ minimizes $L(\boldsymbol{x}, \underline{\boldsymbol{u}}, \underline{t})$, then we must have $g(\underline{\boldsymbol{x}}) \geq 0$ and moreover $\underline{u}_i \cdot g_i(\underline{\boldsymbol{x}}) = 0$ for all $i = 1, 2, \ldots, m$. Thus, $\underline{\boldsymbol{x}}$ is a Kuhn Tucker point. □

Appendix II a)

Problem # 38

Objective function:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 = x_3^2)^2 + (1 - x_3)^2 \\ +10.1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19.8(x_2 - 1)(x_4 - 1)$$

Constraints:

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \ldots, 4.$$

Problem # 43

Objective function:

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

Constraints:

$$8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 \geq 0$$
$$10 - x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 \geq 0$$
$$5 - 2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 \geq 0$$

Problem # 64

Objective function:

$$f(x) = 5x_1 + 50000/x_1 + 20x_2 + 7200/x_2 + 10x_3 + 144000/x_3$$

Constraints:

$$1 - 4/x_1 - 32/x_2 - 120/x_3 \geq 0$$
$$x_i \geq 1 \cdot E - 5, \quad i = 1, 2, 3$$

Problem # 104

Objective function:

$$f(x) \quad = \quad .4x_1^{.67}x_8^{-.67} + .4x_2^{.67}x_8^{-.67} + 10 - x_1 - x_2$$

Constraints:

$$\begin{aligned}
1 - .0588x_5x_7 - .1x_1 &\geq 0 \\
1 - .0588x_6x_8 - .1x_1 - .1x_2 &\geq 0 \\
1 - 4x_3x_5^{-1} - 2x_3^{-.71}x_5^{-1} - .0588x_3^{-1.3}x_7 &\geq 0 \\
1 - 4x_4x_6^{-1} - 2x_4^{-.71}x_6^{-1} - .0588x_4^{-1.3}x_8 &\geq 0 \\
1 \leq f(x) \leq 4.2 & \\
.1 \leq x_i \leq 10, \quad i = 1, \ldots, 8 &
\end{aligned}$$

Problem # 108

Objective function:

$$f(x) \quad = \quad -.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

Constraints:

$$\begin{aligned}
1 - x_3^2 - x_4^2 &\geq 0 & 1 - x_9^2 &\geq 0 \\
1 - x_5^2 - x_6^2 &\geq 0 & 1 - x_1^2 - (x^2 - x_9)^2 &\geq 0 \\
1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 &\geq 0 & & \\
1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 &\geq 0 & & \\
1 - (x_3 - x_5)^2 - (x_4 - x_6)^2 &\geq 0 & & \\
1 - (x_3 - x_7)^2 - (x_4 - x_8)^2 &\geq 0 & & \\
1 - x_7^2 - (x_8 - x_9)^2 &\geq 0 & x_1x_4 - x_2x_3 &\geq 0 \\
x_3x + 9 &\geq 0 & -x_5x_9 &\geq 0 \\
x_5x_8 - x_6x_7 &\geq 0 & 0 &\leq x_9
\end{aligned}$$

Problem # 113

Objective function:

$$
\begin{aligned}
f(x) \quad = \quad & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\
& +4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\
& +7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45
\end{aligned}
$$

Constraints:

$$
\begin{aligned}
105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 &\geq 0 \\
-10x_1 + 8x_2 + 17x_7 - 2x_8 &\geq 0 \\
8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 &\geq 0 \\
-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 &\geq 0 \\
-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 &\geq 0 \\
-.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 &\geq 0 \\
-x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 &\geq 0 \\
3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} &\geq 0
\end{aligned}
$$

A Real-Life Allocation Problem

$$\text{Minimize} \quad -\sum_k^6 D_k g_k(c_k) - \sum_k^6 U_k x_k h_k(q_k) + P\sum_k^6 x_k x_{k+6}$$

$$\text{Subject to} \quad \sum_k^6 x_k x_{k+1} \le 8896$$

$$0 \le x_1 \le 168, \qquad 0 \le x_2 \le 73,$$
$$0 \le x_3 \le 5, \qquad 0 \le x_4 \le 457,$$
$$0 \le x_5 \le 14, \qquad 0 \le x_6 \le 3,$$
$$1 \le x_7 \le 45.2, \qquad 1 \le x_8 \le 54.5,$$
$$1 \le x_9 \le 21, \qquad 1 \le x_{10} \le 6,$$
$$1 \le x_{11} \le 10, \qquad 1 \le x_{12} \le 79,$$

$$\text{where} \quad c_k = \frac{x_k}{D_k}, \qquad q_k = \frac{x_{k+6}}{U_k}$$

$$g_k(c_k) = \frac{-\beta_k}{1 - \underline{C}_k^*}\left(c_k - \frac{c_k^2}{2}\right)$$

$$h_k(q_k) = \frac{0.8P}{\sigma_k}\left(1 - q_k^{-\sigma_k}\right)$$

$$\underline{C}^* = \begin{bmatrix} 0.8085 \\ 0.8816 \\ 0.8778 \\ 0.9375 \\ 0.9615 \\ 0.8 \end{bmatrix} \qquad \sigma = \begin{bmatrix} -0.4 \\ -0.39745095 \\ -0.67372469 \\ 0.15717791 \\ 0.29817787 \\ -0.51789816 \end{bmatrix} \qquad \beta = \begin{bmatrix} 80.19772 \\ 96.68892 \\ 37.85349 \\ 10.50878 \\ 17.48416 \\ 140.92702 \end{bmatrix}$$

$$P = 1.72$$
$$D = (168, 73, 5, 457, 14, 3)^T$$
$$U = (45.2, 54.5, 21.0, 6.0, 10.0, 79.0)^T$$

Appendix II b)

Nonsmooth Problem

Objective function:

$$f(x) = \begin{cases} f_1 = x_2 + 10^{-5}(x_2 = x_1)^2 - 1.0, & \text{if } 0 \leq x_1 < 2 \\ f_2 = \frac{1}{27\sqrt{3}}\left((x_1 - 3)^2 - 9\right) x_2^3, & \text{if } 2 \leq x_1 < 4 \\ f_3 = \frac{1}{3}(x_1 - 2)^3 + x_2 - \frac{11}{3}, & \text{if } 4 < x_1 \leq 6 \end{cases}$$

Subject to:

$$\begin{aligned} g_1 &= \frac{x_1}{\sqrt{3}} - x_2 \geq 0 \\ g_2 &= -x_1 - \sqrt{3}x_2 + 6 \geq 0 \\ & \quad 0 \leq x_1 \leq 6, \quad \text{and} \\ & \quad x_2 \geq 0 \end{aligned}$$

Appendix II c)

Design of multiproduct batch plants

This problem was considered by Kocis and Grossmann (1988), and by Salcedo (1992). A plant consists of M processing stages in series where fixed amounts Q_i of N products have to be manufactured. The objective is to determine for each stage j the number of parallel units N_j and their sizes V_j and for each product i, the corresponding batch sizes B_i and cycle time T_{Li}. The problem data are the horizon time H, the size factors S_{ij} and the processing time t_{ij} of product i i stage j, the required productions Q_i, and appropriate cost coefficients α_j and β_j.

The mathematical formulation of the optimum design plant is as follows.

$$\text{Minimize } \sum_{j=1}^{M} \alpha_j N_j V_j^{\beta_j}$$

subject to

$$\sum_{i=1}^{N} Q_i \frac{T_{Li}}{B_i} \leq H$$

$$V_j \geq S_{ij} B_i$$

$$N_j T_{Li}^1 \geq t_{ij}$$

$$1 \leq N_j \leq N_j^u$$

$$V_j^1 \leq V_j \leq V_j^u$$

$$T_{Li}^1 \leq T_{Li} \leq T_{Li}^u$$

$$B_i^1 \leq B_i \leq B_i^u$$

$$N_j \text{ integers}$$

The bounds N_j^u, V_j^1 and V_j^u are specified by the problem and the appropriate bounds for T_{Li} and B_i can be determined as follows:

$$T_{Li}^1 = \max_j(t_{ij}/N_j^u), T_{Li}^u = \max_j(t_{ij}), b_i^1 = (Q_i/H)T_{Li}^1, B_i^u = \min\{Q_i, \min_j(V_j^u/S_{ij}\}.$$

Input data for the problem are given by:

$M = 6, N = 5$ and $\alpha_j = 250, \beta_j = 0.6, N_j^u = 4, V_j^1 = 300, V_j^u = 3,000$ for $j = 1, 2, 3, \ldots, 6$.

$$Q_1 = 250,000, Q_2 = 150,000, Q_3 = 180,000, Q_4 = 160,000, Q_5 = 120,000.$$

$$S_{ij} = \begin{bmatrix} 7.9 & 7.0 & 5.2 & 4.9 & 6.1 & 4.2 \\ 0.7 & 0.8 & 0.9 & 3.4 & 2.1 & 2.5 \\ 0.7 & 2.6 & 1.6 & 3.6 & 3.2 & 2.9 \\ 4.7 & 2.6 & 1.6 & 2.7 & 1.2 & 2.5 \\ 1.2 & 3.6 & 2.4 & 4.5 & 1.6 & 2.1 \end{bmatrix} \qquad T_{ij} = \begin{bmatrix} 6.4 & 4.7 & 8.3 & 3.9 & 2.1 & 1.2 \\ 6.8 & 6.4 & 7.5 & 4.4 & 2.3 & 3.2 \\ 1.0 & 6.3 & 5.4 & 11.9 & 5.7 & 6.2 \\ 3.2 & 3.0 & 3.5 & 3.3 & 2.8 & 3.4 \\ 2.1 & 2.5 & 4.2 & 3.6 & 3.7 & 2.2 \end{bmatrix}$$