Journal of Global Optimization **29:** 113–120, 2004.

An Exact Method for Fractional Goal Programming

CHARLES AUDET[1], EMILIO CARRIZOSA[2] and PIERRE HANSEN[3]
[1]*Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada (e-mail: charles.audet@gerad.ca, www.gerad.ca/Charles.Audet)*
[2]*Facultad de Matemáticas, Universidad de Sevilla, Tarfia s/n, 41012 Sevilla, Spain (e-mail: ecarrizosa@us.es)*
[3]*GERAD and Department of Quantitative Methods in Management, École des Hautes Études Commerciales, 3000 chemin de la Côte-Saint-Catherine, Montréal (Québec), H3T 2A7 Canada (e-mail: pierreh@crt.umontreal.ca)*

(Received: 13 June 2001; revised: 13 October 2003; accepted: 4 January 2004)

Abstract. Goal Programming with fractional objectives can be reduced to mathematical programming with a linear objective under linear and quadratic constraints, thus optimal solutions can be obtained by using existing Global Optimization techniques. However, only heuristic procedures are suggested in the literature on the field. In this note we explore the practical applicability of a recent algorithm for nonconvex quadratic programming with quadratic constraints for this problem. Encouraging computational experiences for randomly generated instances with up to 14 fractional objectives are presented.

Key words: fractional programming, goal programming, nonconvex optimization, quadratic programming.

Fractional Goal Programming [6, 11, 12] can be expressed as follows:

$$\min_{x\in X}\sum_{j=1}^{m}\left[w_j^+\max\left(\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}-t_j,0\right)+w_j^-\max\left(t_j-\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j},0\right)\right] \tag{1}$$

where $X\subset\mathbb{R}^n$ is a given bounded polyhedron, $w_j^+, w_j^-, t_j\in\mathbb{R}_+$; $\alpha_j,\lambda_j\in\mathbb{R}^n$ and $\beta_j,\mu_j\in\mathbb{R}$ are such that $0<\lambda_j^T x+\mu_j$ for all $x\in X$, for all $j=1,2,\ldots,m$.

In words, one seeks a feasible solution $x\in X$ which minimizes the weighted sum of deviations of m fractional objectives $(\alpha_j^T x+\beta_j)/(\lambda_j^T x+\mu_j)$ from their target values j_j. As pointed out in [8] this model has important applications in areas such as finance, resource allocation, education and others. See also the Appendix of [11] for several further references.

In some cases the optimal value of Problem (1) is 0; this indicates that all fractional objectives are at their target value. This can be verified quickly by finding a point (e.g., by solving a linear program) on the polyhedron

$$\{x\in X:(\alpha_j^T-t_j\lambda_j^T)x=t_j\mu_j-\beta_j \quad \text{for } j=1,2,\ldots,m\}.$$

Problem (1) has a nonconvex objective. To the best of our knowledge, solution methods proposed to date for the general case are all heuristic. They consist

either of finding a local optimum with a nonlinear programming algorithm or of enumerating implicitly vertices of X to find the best solution among them [7, 8]. However, (1) may have no optimal solution at such a vertex as shown by the following example:

$$\min_x \left[\max\left(\frac{x_1}{x_2}-1,0\right)+\max\left(1-\frac{x_1}{x_2},0\right)\right]$$

subject to $x\in X=\{(x_1,x_2): x_1+x_2=4, x_1\geqslant 1, x_2\geqslant 1\}$ where the unique optimal solution $x_1^*=x_2^*=2$ is of value 0, and not at a vertex of X.

However, a weaker localization result can be obtained:

PROPOSITION 1. *There exists an optimal solution x^* to Problem* (1) *and a face F of X, with dimension* $\dim(F)$, *such that $x^*\in F$ and the number of indices j such that $(\alpha_j^T x+\beta_j)/(\lambda_j^T x+\mu_j)=t_j$ is at least* $\dim(F)+1-m$.

Proof. The strategy of the proof is similar to that given in [10] to obtain localization results for a different nonlinear fractional problem.

Since the feasible region of Problem (1) is assumed to be bounded, and all the fractions have strictly positive denominators on X, an optimal solution y exists. The result is straightforward if $\dim(X)\leqslant m-1$, since we can take $F=X, x^*=y$. Hence we assume that $\dim(X)\geqslant m$.

For any $x\in X$, define the index sets $J^{\geqslant}(x)$, $J^{\leqslant}(x)$ as

$$J^{\geqslant}(x)=\left\{j: \frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}\geqslant t_j\right\} \quad \text{and} \quad J^{\leqslant}(x)=\left\{j: \frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}\leqslant t_j\right\}.$$

Consider the auxiliary problem

$$\min_{x\in X}\sum_{j=1}^{m}\left[w_j^+ \max\left(\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}-t_j,0\right)+w_j^- \max\left(t_j-\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j},0\right)\right]$$

subject to:

$$\begin{aligned}
&\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}\geqslant t_j \quad \forall j\in J^{\geqslant}(y)\\
&\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}\leqslant t_j \quad \forall j\in J^{\leqslant}(y).
\end{aligned} \tag{2}$$

Clearly, any optimal solution to the auxiliary problem (2) is also optimal for Problem (1). In particular, y is one such optimal solution.

Then, by Corollary 23 of [2], there exists a face F_0 of the polyhedron defined by the constraints defining problem (2), containing an optimal solution x^* to Problem (2) – and also to Problem (1) – with dimension at most $m-1$.

Moreover, F_0 can be expressed as the intersection of a face F of X and the linear space

$$\left\{\frac{\alpha_j^T x+\beta_j}{\lambda_j^T x+\mu_j}=t_j \ \forall j\in J^{\geqslant}(x^*)\cap J^{\leqslant}(x^*)\right\}.$$

Hence, the cardinality of $J^{\geqslant}(x^*) \cap J^{\leqslant}(x^*)$ must be at least $\dim(F) - m + 1$, as asserted. □

Problem (1) is a structured global optimization one that can easily be reduced to another problem for which solution methods are readily available. Namely, we will reduce it to an instance of a nonconvex quadratic programming problem with quadratic constraints that has a linear objective, and will apply the recent algorithm of [1] (we will refer to this algorithm by QP).

Let d_j^+ and d_j^- represent the deviations from the targets t_j from above and below, and v_j the denominators of the corresponding fractional objective functions. Then easy manipulations show that (1) is equivalent to

$$\min_{x \in X, v, d^+, d^-} \sum_{j=1}^{m} (w_j^+ d_j^+ + w_j^- d_j^-)$$

subject to:

$$\left.\begin{array}{r} d_j^+ v_j + t_j v_j - \alpha_j^T x \geqslant \beta_j \\ d_j^- v_j - t_j v_j + \alpha_j^T x \geqslant -\beta_j \\ v_j - \lambda_j^T x = \mu_j \\ d_j^+, d_j^- \geqslant 0 \end{array}\right\} \quad \text{for } j = 1, \ldots, m. \tag{3}$$

The complementarity constraints

$$d_j^+ d_j^- = 0 \quad (\text{or } \leqslant 0) \tag{4}$$

also hold, and will be automatically satisfied at the optimum, due to the non-negativity of weights w_j^+, w_j^- and deviations d_j^+ and d_j^-. This is a mathematical program with linear objective and with linear and nonconvex quadratic constraints. Moreover, its feasible region is bounded since, by assumption X is bounded, thus finite lower and upper bounds on each variable x and on the denominators v can be obtained, and moreover the deviations satisfy

$$0 \leqslant d_k^+ \leqslant \max\left\{0, \max_j \left(\max_{x \in X} \frac{\alpha_j^T x + \beta_j}{\lambda_j^T x + \mu_j} - t_j \right)\right\}$$

$$0 \leqslant d_k^- \leqslant \max\left\{0, \max_j \left(t_j - \min_{x \in X} \frac{\alpha_j^T x + \beta_j}{\lambda_j^T x + \mu_j} \right)\right\}.$$

Hence, upper bounds can be obtained after solving a series of linear programs.

The algorithm QP solves any quadratically constrained quadratic problem with bounded variables through a branch and cut enumeration scheme. The algorithm can be easily modified to take into account the complementarity information presented above. We will denote by FRAC the same algorithm as QP except that when branching forces $d_j^- \geqslant C$ (for some $C > 0$), then the additional constraint $d_j^+ = 0$ is imposed, and similarly, when branching forces $d_j^+ \geqslant C$ (for some $C > 0$), then the additional constraint $d_j^- = 0$ is imposed.

We solve a series of randomly generated instances of increasing size with both algorithms QP and FRAC. They are generated as follows. The variables x_i are constrained to be in the interval [0,1] (for $i=1,2,\ldots,n$). Every coefficient is rounded to the second decimal. The components of the weights vectors w^- and w^+ are randomly chosen between 0 and 1. For the denominator, λ_j is a random vector with components between -1 and 1; if λ_{ji} is nonnegative, then μ_j is a random number between 1/100 and 1 and if λ_{ji} is negative then μ_j is the sum of λ_{ji} with a random number between 1/100 and 1 (for $j=1,2,\ldots,m$ and $i=1,2,\ldots,n$). For the numerator, the following steps are repeated until the target vector t is positive: let α_j and β be random vectors with components between -1 and 1, let x_j be a random vector with components between 0 and 1; set $t_j=(\alpha_j^T x_j+\beta_j)/(\lambda_j^T x_j+\mu_j)$ (for $j=1,2,\ldots,m$). Only instances for which the optimal objective value is non-zero are kept.

This way of generating the instances ensures that there exists an $x\in X$ that meets each target (of course, the x varies for each target). Moreover, this construction ensures that the assumption that $0<\lambda_j^T x+\mu_j$ for all $x\in X$ is satisfied.

Numerical experiments are done using the QP implementation of [1], and slightly modified for FRAC (the code is designed for instances where the number of variables is small and is written in C++ with CPLEX 7.5 and executed on a SPARC SUN-BLADE-100 under Solaris 2.8). The entries in the following tables are the means (μ) and standard deviations (σ) of the computational times in seconds for thirty randomly generated problems.

Table 1 presents the statistics for eight different runs for each combination of $m\in\{6,10,14\}$ and $n\in\{5,10,15\}$. The columns are partitioned in instances where the redundant constraint appearing in Equation (4) does not explicitly appear in the model formulation, and those where it appears (the rationale for including this complementarity constraint is that its presence reduces the feasible region of the domain of the continuous relaxation used by the algorithm). Given a solution produced by a heuristic method, the algorithm can be used to either confirm its global optimality or to show that it is not optimal by finding the true global solution. In order to simulate the use of a good heuristic method, the algorithm was executed twice: Once to obtain the optimal solution, and a second time only to show that the solution in indeed optimal. The columns of the table are also partitioned into runs without and with a heuristic solution. Finally, the rows indicate which of the two algorithms was used.

Table 1 indicates that when $m=10$ or 14, the use of an heuristic method reduces significantly the computational time required by the algorithm. The table also suggests that when using the heuristic, the presence of the constraint in Equation (4) slightly increases the computational times. Moreover, when the heuristic is used, and when Equation (4) is used, then computing times of both algorithms are comparable. Computational times do not vary significantly when $m=6$. Table 2 displays more computational results based on these observations.

Table 2 suggests that the computational time increases with the number of targets m, and that it decreases with the number of variables. This is probably due

Table 1. Computationals results with or without Equation (4), with or without a simulated heuristic solution, for two versions of the algorithm.

		Without Equation (4)				With Equation (4)			
		Without heuristic		With heuristic		Without heuristic		With heuristic	
		μ	σ	μ	σ	μ	σ	μ	σ
$m=6$									
$n=5$	QP	11.2	17.7	4.4	3.3	9.0	14.3	5.7	4.4
	FRAC	13.2	24.8	4.2	2.5	8.6	13.0	6.1	5.3
$n=10$	QP	6.7	5.5	6.0	4.6	6.6	5.2	7.1	5.1
	FRAC	6.6	5.3	6.0	4.4	6.5	5.2	7.0	5.1
$n=15$	QP	5.4	5.8	6.3	1.3	5.2	4.5	7.3	2.0
	FRAC	5.4	5.8	6.2	1.2	5.2	4.4	7.3	2.0
$m=10$									
$n=5$	QP	930.5	1499.3	360.7	966.5	865.5	1353.9	380.6	1046.8
	FRAC	936.1	1461.5	349.6	950.1	818.1	1189.0	400.0	1087.7
$n=10$	QP	239.4	487.1	137.9	649.7	204.5	365.8	102.8	434.2
	FRAC	234.1	461.3	89.1	380.6	212.3	417.4	108.7	466.1
$n=15$	QP	21.8	45.2	18.4	9.9	22.7	54.8	22.9	17.6
	FRAC	22.0	46.2	19.0	10.4	23.3	55.7	23.9	19.8
$m=14$									
$n=5$	QP	13033.7	15343.6	2711.3	3946.8	12171.7	13375.6	2777.6	4101.4
	FRAC	13228.4	14627.4	2751.7	4074.6	12004.3	13513.0	2758.6	4175.4
$n=10$	QP	5470.6	7331.6	505.9	698.6	5012.2	7056.4	496.4	656.9
	FRAC	5243.0	6614.8	513.9	736.8	4801.2	6379.4	497.0	660.7
$n=15$	QP	1131.7	3032.7	87.0	120.5	863.6	1887.6	94.1	102.6
	FRAC	1049.2	2816.9	83.5	118.6	873.0	2020.6	88.8	97.9

Table 2. Computational results of FRAC with Equation (4), with or without a simulated heuristic solution.

m	$n=5$				$n=10$				$n=15$			
	Without		With heuristic		Without		With heuristic		Without		With heuristic	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
6	8.6	13.0	6.1	5.3	6.5	5.2	7.0	5.1	5.2	4.4	7.3	2.0
7	39.3	62.8	20.8	35.1	9.6	14.7	7.0	1.7	5.9	4.6	8.8	2.7
8	142.7	314.5	45.1	96.2	31.7	78.2	32.4	72.5	16.0	29.2	15.8	9.8
9	196.6	302.8	35.8	58.0	62.8	114.7	36.0	113.1	19.0	45.1	18.3	13.2
10	818.1	1189.0	400.0	1087.7	212.3	417.4	108.7	466.1	23.3	55.7	23.9	19.8
11	1196.8	1433.2	107.2	155.9	853.4	2395.8	655.4	1819.1	243.3	993.4	60.0	152.3
12	3962.4	5373.0	1291.2	2526.4	1786.3	3449.2	267.2	590.7	324.4	868.1	53.1	81.9
13	8126.9	7603.9	1427.6	2230.3	3292.2	6278.9	683.5	1807.0	323.1	638.7	65.9	65.7
14	12004.3	13513.0	2758.6	4175.4	4801.2	6379.4	497.0	660.7	873.0	2020.6	88.8	97.9

to the fact that when there are much more variables than targets, there is more flexibility in attaining several of the target values. Another evident observation is that the standard deviations of the computational times are high. The reason is that for each fixed values of n and m, some of the randomly generated instances were solved very quickly by the algorithm, and others required a long time.

Remark 1. If $m=1$, separating the cases $d_1^+=0$ and $d_1^-=0$ reduces problem (1) to two fractional linear programs and hence to two linear programs [3, 11]. The case where there are several priority levels and one objective at each level can be treated similarly [8].

Remark 2. If instead of weighting positive and negative deviations by single values, one considers convex piecewise functions of these deviations models (1) and (3)–(4) are readily extended, using more variables, as suggested in [11].

Remark 3. If the numerator of denominator of some or all fractional objectives is quadratic instead of linear, but still satisfies non-negativity constraints, the program generalizing (3) still has a linear objective and quadratic constraints. One application is to chance-constrained goal programming (e.g., [9]). If higher-order terms appear, thery can be reduced to quadratic terms by introducing additional variables and quadratic constraints [5].

Remark 4. If instead of polynomials, the numerators and/or denominators of some or all objectives are posynomials or signomials [4] *with simple ratios as powers*, satisfying positivity constraints, similar reduction techniques apply [5], at least in principle as the growth in number of quadratic terms may render the resolution process lengthly.

For comparison purposes, we present the data of one instance with $m=7$ and $n=5$ together with the optimal solution x^*, v^*, d^{+*}, d^{-*}, the optimal value z^* and the computational times in Table 3.

$$X = \{x \in \mathbb{R}^5 : 0 \leqslant x \leqslant 1\}$$

Table 3. Computational results with or without Equation (4), with or without a simulated heuristic solution, for two versions of the algorithm on an instance with $m=7$ and $n=5$.

Equation (4)	Without		With			Without		With	
Heuristic	Without	With	Without	With		Without	With	Without	With
QP	116	55	119	74	FRAC	123	51	120	60

$$\alpha = \begin{bmatrix} -0.36 & 0.42 & 0.15 & 0.18 & 0.87 \\ -0.29 & 0.72 & 0.72 & -0.82 & 0.21 \\ -0.43 & 0.68 & -0.06 & -0.01 & -0.07 \\ 0.72 & -0.82 & 0.84 & -0.01 & -0.33 \\ 0.17 & 0.78 & -0.38 & 0.14 & 0.37 \\ 0.33 & -0.64 & 0.02 & 0.23 & 0.23 \\ -0.5 & 0.65 & 0.3 & 0.56 & 0.77 \end{bmatrix},$$

$$\lambda = \begin{bmatrix} -0.2 & -0.27 & -0.22 & -0.41 & -0.04 \\ -0.19 & 0.93 & -0.99 & 0.99 & 0.45 \\ 0.82 & -0.91 & 0.28 & 0.37 & 0.24 \\ 0.19 & 0.67 & -0.72 & -0.32 & -0.84 \\ -0.4 & -0.55 & -0.74 & -0.76 & 0.83 \\ 0.91 & 0.89 & -0.51 & -0.94 & -0.62 \\ -0.16 & -0.84 & -0.25 & 0.86 & 0.15 \end{bmatrix},$$

$$w^+ = \begin{bmatrix} 1.39 \\ 1.86 \\ 1.27 \\ 1.19 \\ 1.42 \\ 1.29 \\ 1.81 \end{bmatrix}, \quad w^- = \begin{bmatrix} 1.13 \\ 1.66 \\ 1.12 \\ 1.28 \\ 1.45 \\ 1.37 \\ 1.42 \end{bmatrix}, \quad \beta = \begin{bmatrix} 0.66 \\ 0.33 \\ 0.52 \\ 0.23 \\ 0.45 \\ 0.52 \\ 0.03 \end{bmatrix}, \quad \mu = \begin{bmatrix} 1.18 \\ 1.32 \\ 1.23 \\ 1.91 \\ 3.4 \\ 2.88 \\ 1.87 \end{bmatrix}, \quad t = \begin{bmatrix} 7.24 \\ 0.27 \\ 0.21 \\ 0.06 \\ 0.39 \\ 0.34 \\ 0.52 \end{bmatrix},$$

$$x^* = \begin{bmatrix} 1 \\ 0.731215 \\ 0.430749 \\ 1 \\ 1 \end{bmatrix}, \quad v^* = \begin{bmatrix} 0.237807 \\ 2.823589 \\ 2.115204 \\ 1.119775 \\ 2.349077 \\ 2.661100 \\ 1.998092 \end{bmatrix}, \quad d^{+*} = \begin{bmatrix} 0 \\ 0 \\ 0.017582 \\ 0.272417 \\ 0.264156 \\ 0 \\ 0.212957 \end{bmatrix},$$

$$d^{-*} = \begin{bmatrix} 0 \\ 0.175576 \\ 0 \\ 0 \\ 0 \\ 0.020344 \\ 0 \end{bmatrix}, \quad z^* = 1.42638.$$

Acknowledgments

Work of the first author was supported by FCAR grant NC72792 and NSERC grant 239436-01. Work of the second author was partially supported by grant BFM2002-04525-C02-02, MCYT, Spain.

References

1. Audet, C., Hansen, P., Jaumard, B. and Savard, G. (2000), A branch and cut algorithm for nonconvex quadratically constrained quadratic programming, *Mathematical Programming*, A87, 131–152.
2. Carrizosa, E. and Plastria, F. (2000), Dominators for multiple-objective quasiconvex maximization problems, *Journals of Global Optimization*, 18, 35–58.
3. Charnes, A. and Cooper, W.W. (1962), Programming with linear fractional functionals, *Naval Research Logistics Quarterly*, 9, 181–186.
4. Duffin, R.J., Peterson, E.L. and Zener, C. (1967), *Geometric Programming*, Wiley, New-York.
5. Hansen, P. and Jaumard, B. (1992), Reduction of indefinite quadratic programs to bilinear programs, *Journal of Global Optimization*, 2(1), 41–60.
6. Kornbluth, J.S.H. (1973), A survey of goal programming, *Omega*, 1, 193–205.
7. Kornbluth, J.S.H. (1986), On the use of multiple objective linear programming algorithms to solve problems with fractional objectives, *European Journal of Operational Research*, 23, 78–81.
8. Kornbluth, J.S.H. and Steuer, E.R. (1981), Goal programming with linear fractional criteria, *European Journal of Operational Research*, 8, 58–65.
9. Lee, S.M. and Olson, D.L. (1985), A gradient algorithm for chance constrained non-linear goal programming, *European Journal of Operational Research*, 22, 359–369.
10. Plastria, F. and Carrizosa, E. (2001), Gauge distances and median hyperplanes, *Journal of Optimization Theory and Applications*, 110, 173–182.
11. Romero, C. (1991), Hidden nonlinearities in linear goal programming models, *Handbook of Critical Issues of Goal Programming*, Pergamon, Oxford.
12. Tamiz, M., Jones, D.J. and Romero, C. (1998), Goal programming for decision making: An overview of the current state-of-the-art, *European Journal of Operational Research*, 111, 569–581.