On the paper "Augmented Lagrangian algorithms for solving the continuous nonlinear resource allocation problem"

L. F. Bueno*   G. Haeser†   O. Kolossoski‡

February 3, 2023

Abstract

In the paper *[Torrealba, E.M.R. et al. Augmented Lagrangian algorithms for solving the continuous nonlinear resource allocation problem. EJOR, 299(1) 46–59, 2021]* an augmented Lagrangian algorithm was proposed for resource allocation problems with the intriguing characteristic that instead of solving the box-constrained augmented Lagrangian subproblem, they propose projecting the solution of the unconstrained subproblem onto such box. The paper suggests that a global convergence theorem was proved for such a method, however, this is somewhat contrary to usual augmented Lagrangian theory, as this strategy can fail in solving the augmented Lagrangian subproblems. In this note we show that the proposed method may indeed fail and we pinpoint the inconsistency of the aforementioned paper as they use two different projections: one for obtaining their convergence results and other in their implementation. Regardless of the lack of theory, their strategy works remarkably well in some classes of problems, thus, we propose an hybrid method which uses their idea as a starting point heuristics, switching to a standard augmented Lagrangian method when the heuristics fails in improving the KKT residual of the problem. We provide numerical results showing that this strategy is successful in accelerating the standard method.

Keywords: Non-linear programming, Resource allocation problem, Augmented Lagrangian method.

1 Introduction

The continuous knapsack problem is a well studied problem with applications in several fields such as economics, engineering and computer theory. There are several methods tailored to this class of problems, and due to its particular structure, a solution can be found very quickly in comparison to using a general purpose algorithm. Continuous knapsack

*Institute of Science and Technology, Federal University of São Paulo, São José dos Campos, SP, Brazil. (lfelipebueno@gmail.com)

†Department of Applied Mathematics, University of São Paulo, São Paulo, SP, Brazil. (ghaeser@ime.usp.br)

‡Department of Applied Mathematics, University of São Paulo, São Paulo, SP, Brazil. (OliverKolossoski@gmail.com)

problems may be classified in several different subclasses, with different algorithms and applications, as exploited in [Bretthauer and Shetty (2002)]. In this paper we address the continuous quadratic resource allocation problem.

When the problem is separable, one may consider the pegging method, see [Bretthauer and Shetty (2002)], branch and bound methods as in [Li and Sun (2006)], or Newton type methods such as [Cominetti et al. (2014)]. Lagrange multiplier methods for non-separable problems can be found in [Bretthauer and Shetty (2002), Patriksson and Strömberg (2015)]. In [Torrealba et al. (2021)] an augmented Lagrangian method for non-separable problems was proposed. Namely, they considered the problem

$$\begin{array}{ll} \text{Minimize} & f(x), \\ \text{s.t.} & b^T x = c, \\ & \ell \leq x \leq u, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function and b, ℓ, and u are vectors in $\mathbb{R}^n$ with $c \in \mathbb{R}$. Considering the Powell-Hestenes-Rockafellar augmented Lagrangian function $L(x, \lambda, r) = f(x) + \lambda(b^T x - c) + \frac{r}{2}(b^T x - c)^2$ corresponding to penalization of the equality constraints, a standard augmented Lagrangian method would define a sequence of penalty parameters $\{r^k\}$ and a sequence of approximate Lagrange multipliers $\{\lambda^k\}$ in order to define a sequence of approximate solutions $\{x^k\}$ by means of approximately solving the sequence of subproblems

$$\begin{array}{ll} \text{Minimize} & L(x, \lambda^k, r^k), \\ \text{s.t.} & \ell \leq x \leq u. \end{array} \tag{2}$$

In [Torrealba et al. (2021)], the authors propose defining the sequence $\{x^k\}$ alternatively by

$$x^k = \Pi_{[\ell,u]}(\arg\min_{x \in \mathbb{R}^n} L(x, \lambda^k, r^k)), \tag{3}$$

however, the projection operator $\Pi_{[\ell,u]}(\cdot)$ onto the box $[\ell, u]$ is used in the paper somewhat loosely, as in their experiments they considered it to mean the Euclidean projection, whereas in their convergence results they considered the quadratic case

$$f(x) := \frac{1}{2} x^T P x - a^T x, \tag{4}$$

with $a \in \mathbb{R}^n$ and P a positive definite matrix, and the projection is taken with respect to the so-called P-norm, which is defined by $\|u\|_P := \sqrt{u^T P u}$ for all $u \in \mathbb{R}^n$. When considering the Euclidean projection, the authors were able to present a somewhat simple formula for computing (3), which makes the strategy appealing, while considering the projection with respect to the P-norm is meaningless, since its computation may be as hard as solving the standard subproblem (2). Unfortunately, when x^k is computed with the Euclidean projection, this strategy may fail, as shown in the next example.

Example 1.1 *Consider the problem*

$$\begin{array}{ll} \textit{Minimize} & f(x, y, z) := x^2 - 2xy + 2y^2 + z^2, \\ \textit{s.t.} & z = 0, \\ & (1, 0, -1) \leq (x, y, z) \leq (5, 1, 1). \end{array} \tag{5}$$

The augmented Lagrangian function for this problem is given by

$$L(x,y,z,\lambda,r) := x^2 - 2xy + 2y^2 + z^2 + \lambda z + \frac{r}{2}z^2,$$

whose gradient with respect to (x,y,z) *is given by*

$$\nabla L\,(x,y,z,\lambda,r) = \begin{bmatrix} 2x-2y \\ -2x+4y \\ (2+r)z+\lambda \end{bmatrix}.$$

In order to compute (3)*, one must first solve the system* $\nabla L(x,y,z,\lambda,r)=0$*, which clearly gives* $x=y=0$ *and* $z=-\frac{\lambda}{2+r}$*. Thus, when using the Euclidean projection in* (3)*, one arrives at the point* $(1,0,\max\{-1,\min\{1,z\}\})^T$*. Assuming that the algorithm converges to a feasible point, it can only converge to* $(1,0,0)^T$*, which is not a solution of the problem (notice that* $f(1,\frac{1}{2},0) < f(1,0,0)$*, with* $(1,\frac{1}{2},0)$ *being the actual solution).*

In fact, the direction found by the algorithm in [Torrealba et al. (2021)] using the Euclidean projection may not even be a descent direction, as is shown in the extreme example below, where we start at the solution of the problem (minimizer) and converge in a single iteration to a maximizer instead.

Example 1.2 *Consider the problem*

$$\begin{array}{ll} \textit{Minimize} & f(x,y) := 9x^2/4 - 2xy + y^2/2 + 3x - 2y, \\ \textit{s.t.} & y = 1, \\ & (0,0) \leq (x,y) \leq (1,1). \end{array}$$

One can check that the minimizer for this problem is $(0,1)^T$ *while* $(1,1)^T$ *is the maximizer. Noticing that* $\nabla f(2,6)=0$*, starting the algorithm at the minimizer with* $\lambda=0$ *and a small value of* r*, let us say,* $r=0.0001$*, on the first iteration we compute* $\arg\min_{x\in\mathbb{R}^n} L(x,\lambda,r) = (1.9999, 5.9999)^T$*, which yields by* (3) *using the Euclidean projection that the first iterate is the maximizer* $(1,1)^T$*. Thus the method stops at this point satisfying both stopping criteria in [Torrealba et al. (2021)].*

Examples 1.1 and 1.2 illustrate that defining an augmented Lagrangian iteration with the iterate x^k given by (3) may not work in general using the Euclidean projection. In the examples presented in Section 4.3 of [Torrealba et al. (2021)] they considered quadratic functions (4) with P being a multiple of the identity matrix, which implies that the Euclidean projection coincides with the projection with respect to the P-norm. Thus those numerical results are not compromised. However, for the problems in Section 4.2 of [Torrealba et al. (2021)], a full matrix P is taken from the literature, which does not imply equality of the aforementioned projections. Despite that, surprisingly, numerical convergence to a solution is still achieved. In fact, even though the augmented Lagrangian subproblems are not being solved, the method somehow is able to converge to a solution. We show a particular example of this behavior in the next example:

Example 1.3 *Consider the problem*

$$\begin{array}{cc} \min & \frac{1}{2}x^T P x - a^T x, \\ s.t & x + y = 0, \\ & (0,-1) \leq (x,y) \leq (5,1), \end{array}$$

with $P = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$, $a = (4,-4)^T$.

Applying the algorithm from [Torrealba et al. (2021)] to this problem with $x^0 = (0,1)^T$ *and* $\lambda = 1$*, the sequence* $\{x^k\}$ *generated converges to the actual solution* $(1,-1)^T$*, yet* $\|P_{[\ell,u]}(x^k - \nabla L(x^k, \lambda^k, r^k))\| = 1$ *for every* k*, which means that the subproblems are not being solved. This behavior is maintained even if one starts the algorithm at the true Lagrange multiplier* $\lambda = 2$ *at the solution. This happens because the generated sequence of approximate Lagrange multipliers* $\{\lambda^k\}$ *generated does not approximate the correct value.*

In the next section we propose a hybrid algorithm which uses the iterate of [Torrealba et al. (2021)] unless it starts to fail in solving the original problem. That is, we compute x^k by (3) using the Euclidean projection, and so benefiting from the efficiency of the proposal in [Torrealba et al. (2021)], and then we monitor the progress in solving the original problem in terms of its KKT residual. This can be done by finding suitable approximate Lagrange multipliers associated with x^k by means of a least squares procedure, which we show that has a closed form solution. Once it is detected that an iterate x^k fails in reducing the KKT residual, the method may switch to any other strategy with guaranteed convergence. We show that this hybrid strategy when combined with a standard augmented Lagrangian or with the multiplier search method as in [Bretthauer and Shetty (2002)] outperforms the corresponding standard method and is more robust than the algorithm introduced in [Torrealba et al. (2021)].

2 A Hybrid General Framework

In [Torrealba et al. (2021)] they present the interesting idea that subproblems (2) of the augmented Lagrangian method need not be approximately solved at each iteration in order to guide the method towards convergence to a solution in the case of a continuous knapsack problem (1) with a convex quadratic objective function (4). Instead, they show that by computing an iterate x^k by means of (3) using the projection with respect to the P-norm, the method enjoys global convergence. However, in most cases, computing this projection is intractable. It turns out that by using the Euclidean projection instead, this computation is straightforward. Thus, we will investigate the use of the Euclidean projection in (3) as an heuristic for speeding up the standard augmented Lagrangian method.

Since we do not expect x^k as computed in (3) to solve the corresponding subproblem of the augmented Lagrangian (2), even when the method performs well, we must devise a way of checking whether the iterates computed in this way are successful or not. If not, one should abandon the heuristic and solve (2) with a standard method. We do so by measuring the progress of $x := x^k$ in satisfying the Karush-Kuhn-Tucker (KKT) conditions for the original problem (1), when the objective function is the quadratic (4), which states that x is feasible for (1) and there must exist Lagrange multipliers $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}^n$ such that

$$Px - a + \lambda b + \mu = 0, \tag{6}$$

$$\mu_i \leq 0 \text{ if } x_i = \ell_i,\ \mu_i \geq 0 \text{ if } x_i = u_i, \text{ and } \mu_i = 0 \text{ otherwise, } i = 1, \cdots, n. \tag{7}$$

Here, λ is the Lagrange multiplier with respect to the equality constraint while μ is the Lagrange multiplier with respect to the box constraints $\ell \leq x \leq u$. Since feasibility is already being controlled by the augmented Lagrangian, we shall monitor the KKT-residual of an iterate x^k by means of how well equations (6) and (7) are being satisfied at $x := x^k$ for a suitable choice of λ and μ.

As the multiplier generated by the algorithm presented in [Torrealba et al. (2021)] may not converge to the correct multiplier, we developed an efficient way to check the KKT residue in x^k. More precisely, for $x := x^k$, we shall compute a least squares solution (λ, μ) of (6)-(7) and measure the corresponding residual. The sign constraints on μ are ignored in computing the least squares solution, which is then projected onto the appropriate orthant, while for inactive constraints the multiplier is forced to be zero.

In order to do so, we deal first with (7) by considering the set $I \subseteq \{1, \cdots, n\}$ of active constraints at x^k, that is, the set of indexes i such that $(x^k)_i = \ell_i$ or $(x^k)_i = u_i$ and we define $\mu_i = 0$ for all $i \notin I$. For simplicity of notation, let us assume that $I = \{1, \ldots, q\} \subseteq \{1, \ldots, n\}$. Denoting $v = a - Px^k$, we may write (6) as the linear system

$$\begin{bmatrix} I_q & \bar{b} \\ 0 & \tilde{b} \end{bmatrix} \begin{bmatrix} \bar{\mu} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{v} \\ \tilde{v} \end{bmatrix}, \tag{8}$$

where I_q is the identity matrix of size $q \times q$ and we consider the partition of the vectors $b^T = [\bar{b}^T \quad \tilde{b}^T]$ and $v^T = [\bar{v}^T \quad \tilde{v}^T]$ in their first q components and the remaining ones. The vector $\bar{\mu}$ represents the first q components of μ, being the remaining ones equal to zero.

Since (8) is not expected to have a solution, we compute its least squares solution, whose corresponding normal equation reads as follows:

$$\begin{bmatrix} I_q & \bar{b} \\ \bar{b}^T & \|b\|^2 \end{bmatrix} \begin{bmatrix} \bar{\mu} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{v} \\ b^T v \end{bmatrix}. \tag{9}$$

We can now solve the system by performing an elementary row operation on (9), arriving at the equivalent system

$$\begin{bmatrix} I & \bar{b} \\ 0^T & \|b\|^2 - \|\bar{b}\|^2 \end{bmatrix} \begin{bmatrix} \bar{\mu} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{v} \\ b^T v - \bar{b}^T \bar{v} \end{bmatrix}, \tag{10}$$

which gives

$$\lambda = \frac{\tilde{b}^T \tilde{v}}{\|b\|^2 - \|\bar{b}\|^2} \tag{11}$$

and

$$\overline{\mu} = \overline{v} - \lambda\bar{b}. \tag{12}$$

In order to address the sign constraint in (7), we project $\overline{\mu}$ onto the appropriate orthant. That is, we redefine $\overline{\mu}$ as $\overline{\mu}_i := \max\{0, \overline{\mu}_i\}$ when $(x^k)_i = u_i$, and $\overline{\mu}_i := \min\{0, \overline{\mu}_i\}$ when $(x^k)_i = \ell_i$ for all $i \in I$. Notice that when x^k is a minimizer of (1) with a unique Lagrange multiplier associated, this procedure is capable of computing the multiplier with the correct sign.

Then, after computing λ and μ in this way, we define the KKT-residual ϵ^{k+1} at each iterate x^k by

$$\varepsilon^{k+1} := \|Px^k - a + \lambda b + \mu\|. \tag{13}$$

Notice that when $\|\bar{b}\| = \|b\|$ this process fails and we simply set $\varepsilon^{k+1} := +\infty$. Note that this is a rather peculiar pathological situation, as it could only happen if all variables have values equal to one of their bounds, which would imply that we have $n+1$ active constraints at x^k. Now, to state our hybrid strategy, we evaluate whether the heuristic in computing x^k by (3) using the Euclidean projection is efficient in reducing the KKT-residual at each iteration, that is, if $\varepsilon^{k+1} \leq \gamma\varepsilon^k$ where $\gamma < 1$ is predefined. If so, we continue with the heuristic, otherwise we switch to solving the augmented Lagrangian subproblem (2). The full algorithm with this modification is stated below.

Algorithm 1 Hybrid augmented Lagrangian algorithm for the resource allocation problem

---

Step 0. Choose $\lambda_{min} \in \mathbb{R}$, $\lambda_{max} \in \mathbb{R}$, $\lambda^0 \in [\lambda_{min}, \lambda_{max}]$, $\gamma < 1$, $r_0 > 0$, $\theta \in (0, 1)$, $\beta > 1$, $k_0 \geq 0$, and $\varepsilon > 0$, and set $k := 0$.

Step 1. If $k \leq 1$ or $\varepsilon^k \leq \gamma\varepsilon^{k-1}$, compute the iterate x^k according to the following procedure:

1.1) Find $\bar{x}^k$ as a solution of $\min\{L(x, \lambda_k, r_k) : x \in \mathbb{R}^n\}$.

1.2) Compute using the Euclidean projection:

$$x^k = \Pi_{[\ell,u]}(\bar{x}^k - \nabla_x L(\bar{x}^k, \lambda_k, r_k)).$$

Otherwise, switch to a standard augmented Lagrangian method, that is, from this point onwards, this step consists of finding x^k by approximately solving subproblem (2) by any method of choice.

Step 3. Compute a new approximation of the Lagrange multiplier, according to:

$$\lambda_{k+1} = \max\{\lambda_{min}, \min\{\lambda_{max}, \lambda_k + r_k(b^T x^k - c)\}\}.$$

Step 4. Update the penalty parameter:

$$r_{k+1} = \begin{cases} r_k, & \text{if } k \leq k_0 \quad \textit{or} \quad |h(x^k)| \leq \theta\ |h(x^{k-1})| \\ \beta\ r_k, & \textit{otherwise} \end{cases}$$

where $h(x) = b^T x - c$.

Step 5. Compute the KKT residual: Use the least squares procedure described to compute λ and μ (formulas (11) and (12)), and compute ε^{k+1} by (13).

Step 6. Test the stopping criterion: If $|b^T x^k - c| < \varepsilon$ and $\|x^k - x^{k-1}\| < \varepsilon$, then stop. Otherwise set $k := k + 1$ and return to **Step 1**.

---

3 Numerical Experiments

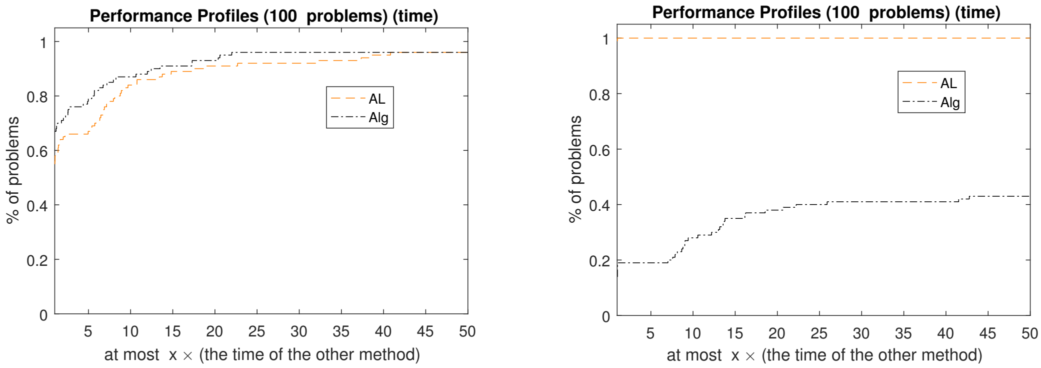
In this section we perform some numerical experiments illustrating the robustness and effectiveness of Algorithm 1 and a variation of it which we describe later. All the experiments were ran with Matlab on an Intel Core i7-8565U 1.99 GHz.

In our first test, we illustrate that the original method in [Torrealba et al. (2021)] may perform well or very poorly depending on the structure of the problem. We select two sets of problems and run the algorithm proposed in [Torrealba et al. (2021)] (abbreviated `Alg`), which corresponds to Algorithm 1 but where **Step 1** is always computed using the heuristic approach (3), never switching to solving subproblem (2). We compare it with a standard augmented Lagrangian approach (abbreviated `AL`), where **Step 1** of Algorithm 1 is replaced by directly solving subproblem (2) at every iteration, using Matlab's optimization toolbox with standard settings.

In each set of problems, we chose 100 randomly generated convex quadratic objective functions with structure defined as in (4). For each problem, entries of vector a were randomly generated in the interval $[0, 1]$, while constraint vector b and the initial point to run each method were taken as e, the vector of ones in $\mathbb{R}^n$, with $n = 500$. Entries of matrix P were randomly generated on $[0, 0.1]$, with P being redefined as $P^T P + I$ in order to ensure positive definiteness. For both methods we used $\varepsilon = 10^{-4}$, $\gamma = 0.9$, and k_0 was chosen large enough so that $r = 1$ was maintained constant, as in the implementation in [Torrealba et al. (2021)]. This also dismisses the choice of the parameters θ and β. Finally, for the first set of 100 problems we considered the constraints defined by the box $[\ell, u] = [-\frac{n}{2}e, \frac{n}{2}e]$ with $c = 0$, while for the second set of 100 problems we considered a displacement of this constraint by considering $[\ell, u] = [0, ne]$ and $c = n$. Finally, in **Step 6**, we considered an additional stopping criterion of a maximum of 1000 iterations for both methods.

In Figure 1a) we show the performance profile of the results on the first set of problems while Figure 1b) shows the correspondent results on the second set of problem. We can see that `Alg` is able to compete with `AL` in the first set of problems, being slightly more efficient and solving almost all problems. However, in the second test, `Alg` performed considerably poorer than `AL`, being able to solve only circa 40% of the problems in less than 50 times the time taken by `AL`.

Figure 1 illustrates that the performance of the algorithm in [Torrealba et al. (2021)] may drastically depend on the structure of the problem, which emphasizes the need of considering the hybrid approach of Algorithm 1 instead. It also suggests that the convergence theory of this method should be further investigated in the sense of detecting larger classes of problems where the algorithm is able to perform well. Notice that in the test



(a) Set 1: $[\ell, u] = [-\frac{n}{2}e, \frac{n}{2}e]$ and $c = 0$.

(b) Set 2: $[\ell, u] = [0, ne]$ and $c = n$.

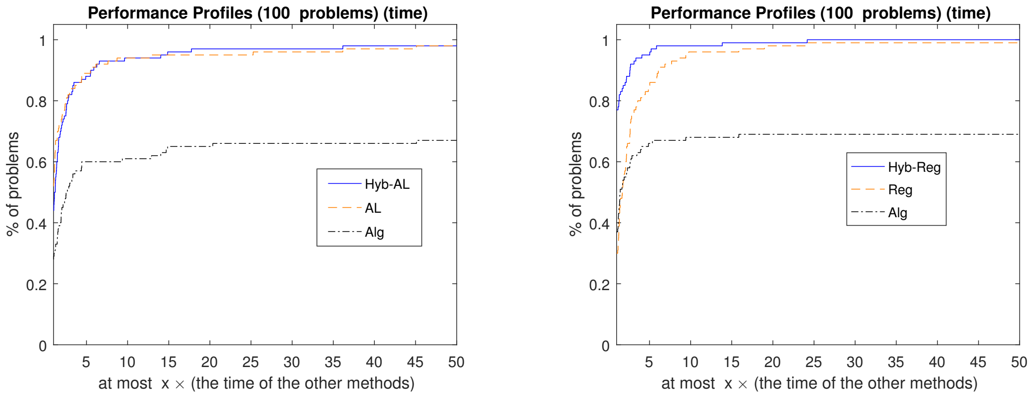
Figure 1: Comparison of the standard augmented Lagrangian (`AL`) and the heuristic approach proposed in [Torrealba et al. (2021)] (`Alg`) on two sets of constraints and 100 randomly generated convex problems.

set depicted in Figura 1a), the method is still able to perform quite well despite the fact that the euclidean projection does not coincide with the projection with respect to the P-norm.

In our second test we considered the behavior of the Hybrid approach we presented in Algorithm 1 (which we abbreviate as `Hyb-AL`) in comparison with the standard augmented Lagrangian approach (`AL`) and the original algorithm in [Torrealba et al. (2021)] (`Alg`). The test set is choosen similarly as before but with a mixture of constraints where `Alg` behaves well or poorly, that is, we considered 50 randomly generated problems with the structure described in Figure 1a) and 50 randomly generated problems as described in Figure 1b). The results are shown in Figure 2a) where we can see that `Hyb-AL` is able to quickly switch to the standard augmented Lagrangian whenever the heuristic is failing without hindering its performance.

Finally, in Figure 2b), we present the results on this same test of problems but considering a different method instead of the augmented Lagrangian approach. That is, we considered Algorithm 1 but in **Step 1**, when the heuristic approach fails in reducing the KKT-residual, we stop the execution and resort to a different method from this point onwards. The method we considered is a version of the multiplier search method as described in [Bretthauer and Shetty (2002)], where the idea is solving the equation $b^T x(\lambda) = c$ using a root finding algorithm, where $x(\lambda)$ is the projection onto the box constraint of the solution x of $Px - a + \lambda b = 0$. We used the Regula Falsi method as the root finding algorithm (see [Bretthauer and Shetty (2002)] for details). We then compare the hybrid algorithm built in this way (`Hyb-Reg`), the original algorithm of [Torrealba et al. (2021)] (`Alg`), and the pure multiplier search method (`Reg`), where the results are shown in Figure 2b). There, we can see that the proposed heuristic is able to accelerate the multiplier search method considerably, being the most efficient method for circa 80% of the problems.

In Figure 2 we can see that the hybrid approach is able to accelerate the method of



(a) Comparison with augmented Lagrangian strategy (b) Comparison with multiplier method using regula falsi

Figure 2: Comparison of the heuristic approach proposed in [Torrealba et al. (2021)] (`Alg`), the Hybrid strategy (`Hyb-AL` or `Hyb-Reg`) based on Algorithm 1, and the respective pure solver (`AL` or `Reg`) on a collection of 100 randomly generated problems from a mixture of constraints from sets 1 and 2 described in Figure 1.

choice, switching to the original method once it detects failure of the heuristic. This is done in such a way that, even when the heuristic fails, the extra work at each step is negligible in comparison with the computation done by the original method. Thus our heuristic approach is able to preserve robustness of the original method while also carrying out the efficiency of the heuristic approach.

4 Conclusions

In this note we highlight an inconsistency in the paper [Torrealba et al. (2021)], where they suggest, instead of solving an augmented Lagrangian subproblem, to first solve an unconstrained version of it and then project its solution onto its corresponding feasible set. The paper shows convergence by means of using the projection with respect to the P-norm, where P is the Hessian of the objective function, however there is no easy way to compute this unless the Euclidean projection is used. In their numerical experiments they considered mostly the case where P is a multiple of the identity, which guarantees that both projections are the same, however the strategy was successful in other instancies as well, which still lacks some explanation.

Thus we suggested an Hybrid strategy combining computation of the iterate using the euclidean projection and a standard augmented Lagrangian method. A simple procedure to measure the success of the computation of the iterate is devised, which gives a criterion for switching to the standard augmented Lagrangian or any other method of choice. In our numerical experiments we show that this strategy was successful in accelerating the standard augmented Lagrangian method and the multiplier search method.

When P is not a multiple of the identity, it remains to be investigated whether one can detect a class of problems where the heuristic approach of [Torrealba et al. (2021)] can be safely used without a safeguarding procedure. We tested that when the box is well centered around the origin (Figure 1a)), the procedure works well despite the inconsistency on the projections, however it is not clear whether one can find a class of problems where the Euclidean projection is different from the projection with respect to the P-norm, but the algorithm may be proved to work.

Acknowledgments

This work was supported by the São Paulo Research Foundation FAPESP grants 2013/07375-0, 2018/24293-0 and 2021/05007-0, and CNPq.

References

[Birgin and Martinez (2014)] Birgin, E. G., Martínez, J. M. (2014). Practical Augmented Lagrangian Methods for Constrained Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA.

[Bretthauer and Shetty (2002)] Bretthauer, K. M., Shetty, B. (2002). The nonlinear knapsack problem-algorithms and applications. European Journal of Operational Research, 138(3) 459-472.

[Cominetti et al. (2014)] Cominetti, R., Mascarenhas, W. F., Silva, P. J. S. (2014). A Newton's method for the continuous quadratic knapsack problem. Mathematical Programming Computation, 6 151-169.

[Li and Sun (2006)] Li, D., Sun, X. (2006). Nonlinear Integer Programming. Springer.

[Patriksson and Strömberg (2015)] Patriksson, M., Strömberg, C. (2015). Algorithms for the continuous nonlinear resource allocation problem: New implementations and numerical studies. European Journal of Operational Research, 243(3) 703-722.

[Torrealba et al. (2021)] Torrealba, E. M. R., Silva. J. G., Matioli. L. C., Kolossoski, O., Santos, P. J. S. (2021). Augmented Lagrangian algorithms for solving the continuous nonlinear resource allocation problem. European Journal of Operational Research, 299(1) 46-59.

[Wright and Lim (2020)] Wright, S. E., Lim, S. (2020). Solving Nest-Constraint Resource Allocation Problems with an Interior Point Method, Operations research Letters, 48 297-303.