Constrained optimization with integer and continuous variables using inexact restoration and projected gradients[*]

E. G. Birgin[†] R. D. Lobato[‡] J. M. Martínez[‡]

September 23, 2016

Abstract

Inexact restoration (IR) is a well established technique for continuous minimization problems with constraints that can be applied to constrained optimization problems with specific structures. When some variables are restricted to be integer, an IR strategy seems to be appropriate. The IR strategy employs a restoration procedure in which one solves a standard nonlinear programming problem and an optimization procedure in which the constraints are linearized and techniques for mixed-integer (linear or quadratic) programming can be employed.

Key words: Inexact restoration, mixed-integer nonlinear programming (MINLP), projected gradients.

1 Introduction

Many practical problems involve the necessity of solving constrained optimization problems with integer and continuous variables (MINLP). See [16, 17, 24], among many others. In the present century, several algorithms for linear or quadratic mixed (continuous and integer) optimization are able to solve problems with hundreds of thousands or even millions of variables and, moreover, several packages for large-scale nonlinear optimization are available (see [10] and the references therein). As a consequence, the idea of designing MINLP algorithms that use the available software for nonlinear programming and mixed-integer linear-quadratic programming is very attractive.

In this paper, we suggest that the inexact-restoration framework provides an adequate scheme for combining mixed-integer (linear or quadratic) programming software and nonlinear programming software in a fruitful way.

[*] This paper has been submitted to the Special Issue of the Bulletin of Computational Applied Mathematics in Honor of the 60th Birthday of Professor Marcos Raydan. This work has been partially supported by the Brazilian agencies FAPESP (grants 2010/10133-0, 2013/03447-6, 2013/05475-7, 2013/07375-0, 2014/18711-3, and 2015/18053-9) and CNPq (grants 309517/2014-1 and 303750/2014-6).

[†] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090 São Paulo, SP - Brazil. E-mail: egbirgin@ime.usp.br

[‡] Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, Campinas, SP, Brazil. E-mails: lobato@ime.usp.br, martinez@ime.unicamp.br

This paper is organized as follows. In Section 2, we present an inexact restoration algorithm for solving MINLP problems. In Section 3, we discuss some implementation details. In Section 4, we present numerical examples to illustrate the application of the proposed method. Finally, we draw some conclusions in Section 5.

Notation. The symbol $\|\cdot\|$ denotes the Euclidean norm of vectors and matrices. Given $a \in \mathbb{R}^n$, we define a_+ as the n-dimensional vector whose i-th component is $\max\{a_i, 0\}$ and a_- as the n-dimensional vector whose i-th component is $\min\{a_i, 0\}$.

2 Algorithm

The problem considered in this paper is

$$\text{Minimize } f(x, y) \tag{1}$$

subject to

$$h(x, y) = 0, \ g(x, y) \leq 0, \tag{2}$$

$$x \in X, \ y \in Y, \ y_i \text{ integer for all } i, \ X \text{ and } Y \text{ are boxes}, \tag{3}$$

x and y have n_x and n_y components, respectively, $f : \mathbb{R}^{n_x+n_y} \to \mathbb{R}$, $h : \mathbb{R}^{n_x+n_y} \to \mathbb{R}^m$, $g : \mathbb{R}^{n_x+n_y} \to \mathbb{R}^p$, and all the functions are sufficiently differentiable.

From now on, we define

$$H(x, y) = \|h(x, y)\| + \|g(x, y)_+\|,$$

and

$$\Phi(x, y, \theta) = \theta f(x, y) + (1 - \theta) H(x, y).$$

As mentioned in [8], the idea of IR methods [3, 6, 9, 13, 14, 15, 23, 26, 27, 28, 29] is that, at each iteration, feasibility and optimality are addressed in different phases. In the Restoration Phase the algorithms aim to improve feasibility and in the Optimization Phase they aim to improve optimality, preserving a linear approximation of feasibility. An IR method that applies to problem (1) is described below.

Algorithm 2.1.

Let $x^0 \in X$, $y^0 \in Y \cap \mathbb{Z}^{n_y}$, $r \in (0, 1)$, $\beta > 0$, $\sigma_0 \geq 0$, and $\theta_0 \in (0, 1)$ be given. Set $k \leftarrow 0$.

Step 1. (Restoration)

Compute $x^k_{\text{rest}} \in X$ and $y^k_{\text{rest}} \in Y \cap \mathbb{Z}^{n_y}$ such that

$$H(x^k_{\text{rest}}, y^k_{\text{rest}}) \leq r H(x^k, y^k) \tag{4}$$

and

$$f(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}) \le f(x^k, y^k) + \beta H(x^k, y^k). \tag{5}$$

Step 2. (Updating the Penalty Parameter θ)

If

$$\Phi(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}, \theta_k) \le \Phi(x^k, y^k, \theta_k) + \frac{1-r}{2}\left(H(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}) - H(x^k, y^k)\right), \tag{6}$$

set $\theta_{k+1} = \theta_k$. Otherwise, set

$$\theta_{k+1} = \frac{(1+r)\left(H(x^k, y^k) - H(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})\right)}{2\left(f(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}) - f(x^k, y^k) + H(x^k, y^k) - H(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}))\right)}. \tag{7}$$

Step 3. (Optimization Phase)

Solve the following mixed-integer (linear or quadratic) optimization problem:

$$\text{Minimize } f'_x(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(x - x^k_{\mathrm{rest}}) + f'_y(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(y - y^k_{\mathrm{rest}}) + \sigma_k \|(x, y) - (x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})\|^2 \tag{8}$$

subject to

$$h'_x(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(x - x^k_{\mathrm{rest}}) + h'_y(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(y - y^k_{\mathrm{rest}}) = 0, \tag{9}$$

$$g'_x(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(x - x^k_{\mathrm{rest}}) + g'_y(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})(y - y^k_{\mathrm{rest}}) + g(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})_- \le 0, \tag{10}$$

$$x \in X \text{ and } y \in Y \cap \mathbb{Z}^{n_y}, \tag{11}$$

obtaining x_{trial} and y_{trial}.

Step 4. (Acceptance or Rejection of the Trial Step)

Test the conditions

$$f(x_{\mathrm{trial}}, y_{\mathrm{trial}}) \le f(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}}) - \sigma_k(\|x_{\mathrm{trial}} - x^k_{\mathrm{rest}}\|^2 + \|y_{\mathrm{trial}} - y^k_{\mathrm{rest}}\|^2) \tag{12}$$

and

$$\Phi(x_{\mathrm{trial}}, y_{\mathrm{trial}}, \theta_{k+1}) \le \Phi(x^k, y^k, \theta_{k+1}) + \frac{1}{2}(1-r)(\|H(x^k_{\mathrm{rest}}, y^k_{\mathrm{rest}})\| - \|H(x^k, y^k)\|). \tag{13}$$

If both (12) and (13) are fulfilled, define $x^{k+1} = x_{\mathrm{trial}}$, $y^{k+1} = y_{\mathrm{trial}}$, update $k \leftarrow k+1$, set $\sigma_{k+1} = 0$, and go to Step 1. Otherwise, choose a new value for σ_k between $1 + 10\sigma_k$ and $1 + 1000\sigma_k$, and go to Step 3.

3 Practical implementation

In order to take advantage of available consolidated software, the Restoration Step will be implemented in such a way that, for a fixed y^k, the point $x^k_{\rm rest}$ so far computed should be the result of applying a robust nonlinear programming algorithm to the problem

$$\text{Minimize } f(x, y^k) \text{ subject to } h(x, y^k) = 0, g(x, y^k) \leq 0, \tag{14}$$

and $x \in X$. Following this idea, we will define $y^k_{\rm rest} = y^k$. Moreover, the conditions (4) and (5) may be used as stopping criteria for the nonlinear programming solver.

Let us consider now the Optimization Step. Observe that the minimizer of

$$f(x^k_{\rm rest}, y^k_{\rm rest}) + f'_x(x^k_{\rm rest}, y^k_{\rm rest})(x - x^k_{\rm rest}) + f'_y(x^k_{\rm rest}, y^k_{\rm rest})(y - y^k_{\rm rest}) + \sigma_k[\|x - x^k_{\rm rest}\|^2 + \|y - y^k_{\rm rest}\|^2]$$

for (x, y) in a polytope is the projection of $(x^k_{\rm rest}, y^k_{\rm rest}) - \frac{1}{2\sigma_k}\nabla f(x^k_{\rm rest}, y^k_{\rm rest})$ onto the polytope. Of course, with the constraint that y should be integer, this statement is not really true. However, because of the observation above, the procedure in the Optimization Phase can be interpreted as a projected gradient integer step. Well established results about the Spectral Projected Gradient method recommend to compute σ_k using the spectral choices suggested in [2, 4, 7, 11, 12, 13, 19, 31, 32] and many others. For solving the mixed-integer (linear or quadratic) nonlinear optimization subproblem we may use CPLEX [25].

4 Numerical examples

To illustrate the method proposed in this work, we consider two mixed-integer nonlinear programming problems: a circle packing problem and the traveling salesman problem with neighborhoods.

4.1 Circle packing

The circle packing problem can be defined as follows. We are given a circle $\mathcal{C}$ (which we call the container) with radius R, centered at the origin, and N circles (which we call the items) with radii $r_1, \dots, r_N$. We assume that $r_i \leq R$ for each $i \in \{1, \dots, N\}$. The problem consists in selecting a subset of the given items. The selected items must be arranged inside the container and must not overlap with each other. The objective is to maximize the sum of the areas of the selected items.

We can model this problem as a mixed-integer nonlinear programming problem. The variables of the model will be $y_i \in \{0, 1\}$, $i \in \{1, \dots, N\}$, that define which items will be selected ($y_i = 1$ if and only if item i is selected), and $c_i \in \mathbb{R}^2$, $i \in \{1, \dots, N\}$, which determine the centers of the items (in case they are selected). The objective is therefore to minimize

$$-\sum_{i=1}^{N} y_i r_i^2 \tag{15}$$

subject to the conditions that each selected item must be contained in $\mathcal{C}$

$$\|c_i\|_2^2 \leq (R - r_i)^2, \quad \forall i \in \{1, \dots, N\} \tag{16}$$

and that the selected items must not overlap with each other

$$\|c_i - c_j\|_2^2 \geq (r_i + r_j)^2(y_i + y_j - 1), \quad \forall i, j \in \{1, \ldots, N\} \text{ such that } i < j. \tag{17}$$

If both y_i and y_j are equal to 1, then constraint (17) becomes $\|c_i - c_j\|_2^2 \geq (r_i + r_j)^2$, which means that the circles with indices i and j must not overlap with each other. Otherwise, if $y_i = 0$ or $y_j = 0$, then the right-hand side of constraint (17) is a nonpositive number, in which case (17) represents no restraint, imposing no condition on c_i and c_j.

We implemented model (15)–(17) and Algorithm 2.1 in AMPL [18]. In the Restoration Step, in order to compute x^k_{rest} and y^k_{rest} that satisfy (4) and (5), we solve problem (14). This is a nonlinear programming problem obtained by fixing the integer variables of the original problem to y^k. In our packing problem, the integer variables determine which items are selected. Since the objective function (15) depends only on the integer variables, problem (14) becomes a feasibility problem. More specifically, it is the problem of arranging the selected items within the container, i.e., finding the center of the items such that (16) and (17) are satisfied. To solve problem (14), we use Algencan [1, 10] version 3.0.0, which is available for downloading at the TANGO Project web page (`http://www.ime.usp.br/~egbirgin/tango/`). The Optimization Phase is responsible for selecting the items. We use AMPL/CPLEX version 12.6.3.0 to solve problem (8)–(11).

Making a parallel with the notation used in the description of Algorithm 2.1, we have $x^k = (c_1^k, \ldots, c_N^k) \in \mathbb{R}^{2N}$ as the sequence of continuous iterates. Initially, we set $y_1^0 = 1$ and $y_i^0 = 0$ for all $i \in \{2, \ldots, N\}$, and c_i^0 is chosen uniformly random in $[-R, R]^2$ for all $i \in \{1, \ldots, N\}$. We have chosen $r = 0.5$, $\beta = 1$, $\sigma_0 = 1$, and $\theta_0 = 0.1$. In our implementation of the Algorithm 2.1, we keep track of the best (feasible) solution found so far. We considered two instances with $N = 10$ items. In the first instance, the radius of the container is $R = 1$ and the radius of the i-th item is $r_i = 0.05i$ for $i \in \{1, \ldots, N\}$. In the second instance, the radius of the container is also $R = 1$ and the radius of the i-th item is $r_i = 0.5i^{-2/5}$ for $i \in \{1, \ldots, N\}$.

Table 1 shows the result we have obtained for the first instance. The first column shows the number of the iteration where the current best solution was updated. The second column displays the items that were selected at that solution. The third column presents the objective function value for that solution. For each iteration before iteration 16 that is not shown in the first column of the table, the set of selected items at that iteration is the same as the set chosen at the previous iteration. Figures 1(a)–(f) depict graphical representations of the solutions found at iterations 3, 4, 8, 9, 12, and 16, respectively. The number displayed within each circle indicates the index of the circle. The results for the second instance are presented in Table 2. Figures 2(a)–(c) exhibit the solutions found at iterations 8, 19, and 69, respectively.

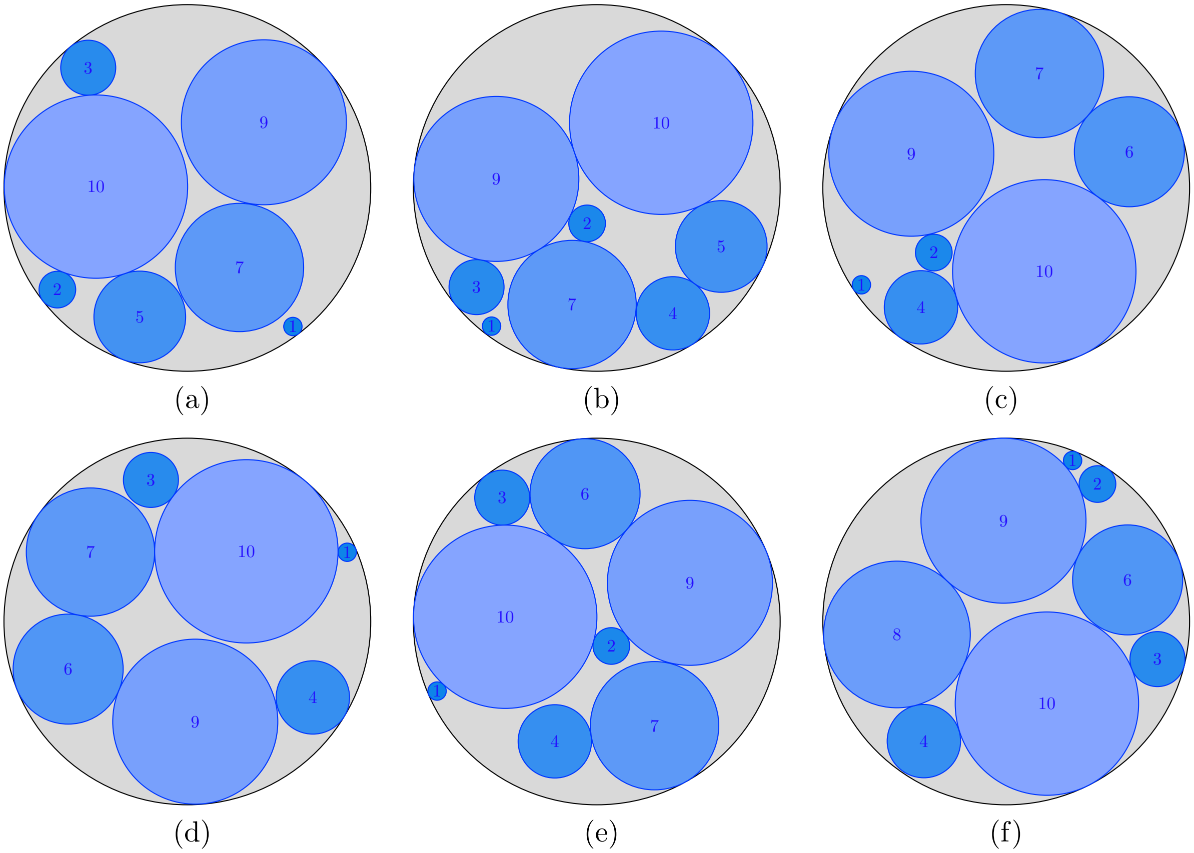
4.2 Traveling salesman problem with neighborhoods

The traveling salesman problem with neighborhoods [5] is an extension of the classical traveling salesman problem. We are given N regions (neighborhoods) and the objective is to find the shortest route that visits each region and returns to the initial departure point. We say that a route visits a region if it passes through a point that belongs to that region.

We can formulate the problem as follows. Let $V = \{1, \ldots, N\}$ be the set of indices of the regions and $E = \{(i, j) \in V^2 \mid i < j\}$. Let $\mathcal{R}_i \subseteq \mathbb{R}^n$ for each $i \in V$ be the given regions.

Iteration	Selected items	Solution value
1	1	-0.0025
3	1, 2, 3, 5, 7, 9, 10	-0.6725
4	1, 2, 3, 4, 5, 7, 9, 10	-0.7125
8	1, 2, 4, 6, 7, 9, 10	-0.7175
9	1, 3, 4, 6, 7, 9, 10	-0.73
12	1, 2, 3, 4, 6, 7, 9, 10	-0.74
16	1, 2, 3, 4, 6, 8, 9, 10	-0.7775

Table 1: Sequence of selected items in the current best solutions found by the algorithm for the first instance, where $r_i = 0.05i$ for $i \in \{1, \dots, N\}$.



(a) (b) (c)

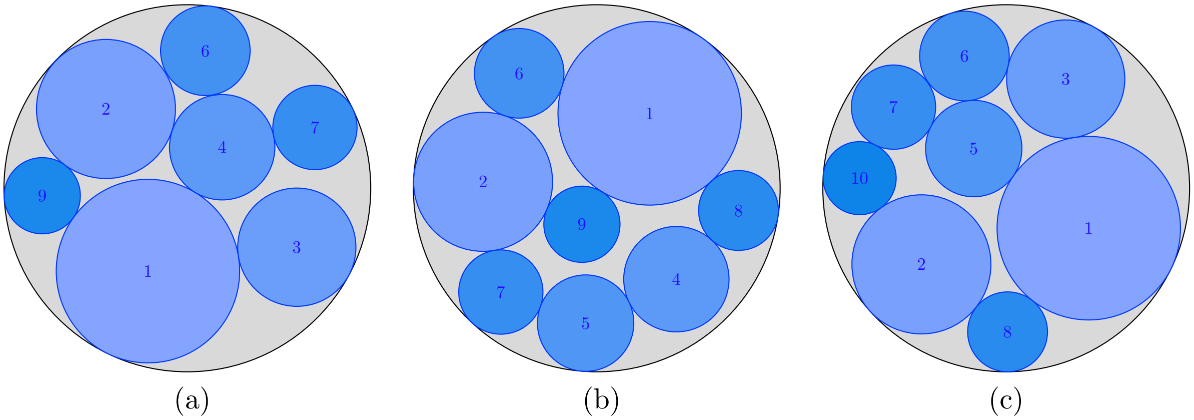
(d) (e) (f)

Figure 1: Illustration of the current best solution found by the algorithm at iterations 3, 4, 8, 9, 12, and 16 for the first instance, where $r_i = 0.05i$ for $i \in \{1, \dots, N\}$.

For each $i \in V$, we let $x_i \in \mathbb{R}^n$ be a variable representing a point in $\mathcal{R}_i$ where the route must pass through. For each $(i, j) \in E$, let y_{ij} be a binary variable that indicates whether the route goes from point x_i to x_j (or from point x_j to x_i), i.e., there is an edge connecting x_i and x_j.

Iteration	Selected items	Solution value
1	1	-0.03962
8	1, 2, 3, 4, 6, 7, 9,	-0.73530
19	1, 2, 4, 5, 6, 7, 8, 9,	-0.74784
69	1, 2, 3, 5, 6, 7, 8, 10	-0.76570

Table 2: Sequence of selected items in the current best solutions found by the algorithm for the second instance, where $r_i = 0.5 i^{-2/5}$ for $i \in \{1, \ldots, N\}$.



(a) (b) (c)

Figure 2: Illustration of the current best solution found by the algorithm at iterations 8, 19, and 69 for the second instance, where $r_i = 0.5 i^{-2/5}$ for $i \in \{1, \ldots, N\}$.

Considering the Euclidean distance between the points, the objective is to minimize

$$\sum_{(i,j)\in E} y_{ij} \|x_i - x_j\|_2. \tag{18}$$

We consider that each region is an ellipsoid, so that $x_i \in \mathcal{R}_i$ for each $i \in V$ can be formulated as

$$(x_i - c_i)^\top M_i (x_i - c_i) \leq 1, \text{ for each } i \in V, \tag{19}$$

where $c_i \in \mathbb{R}^n$ is the center of the ellipsoid and $M_i \in \mathbb{R}^{n \times n}$ is symmetric and definite positive. Since the route must goes into and out each region, we must have

$$\sum_{j=1}^{i-1} y_{ji} + \sum_{j=i+1}^{N} y_{ij} = 2, \text{ for each } i \in V. \tag{20}$$

Finally, to guarantee that the route connects all regions, there must be a path that connects any subset of the regions to a region outside of that subset:

$$\sum_{i\in S} \left(\sum_{\substack{j\in S\\ j<i}} y_{ji} + \sum_{\substack{j\in S\\ j>i}} y_{ij} \right) \leq |S| - 1, \text{ for each } S \subseteq V, |S| \geq 3. \tag{21}$$

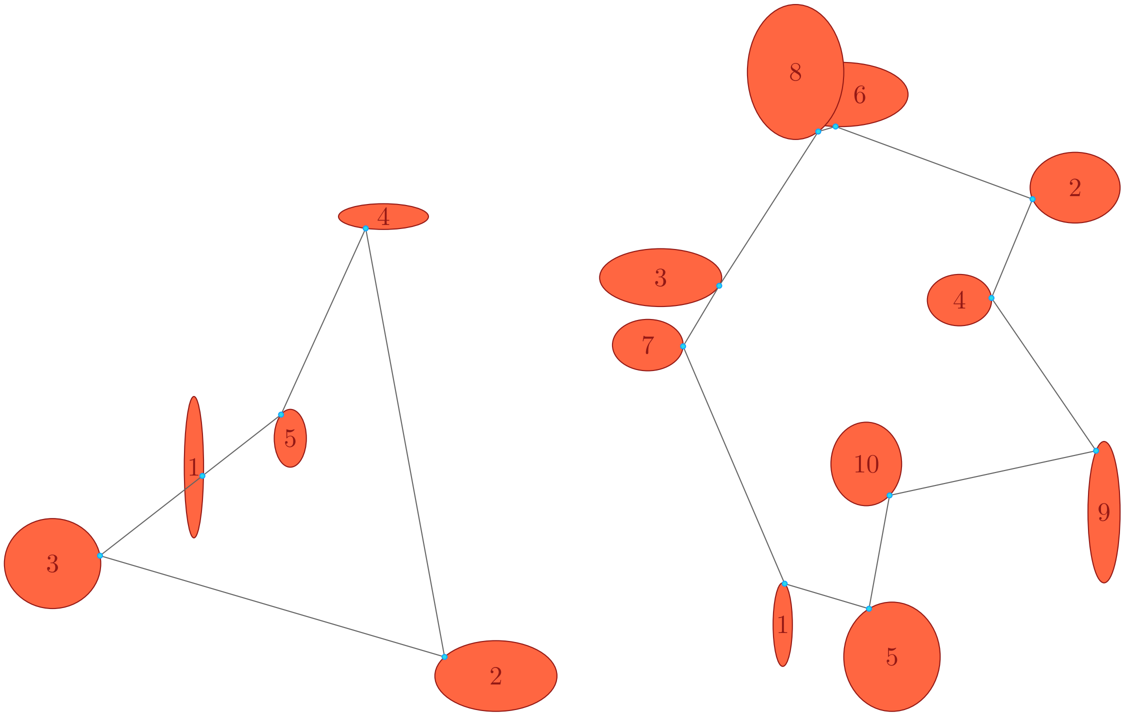
Instance	Iteration	Solution value
tspn2DE5_1	1	213.11675
	2	191.25520
tspn2DE10_1	1	469.48924
	2	225.12607
tspn2DE15_1	1	646.72179
	2	345.52009
	3	291.23176
	39	290.41621
	53	290.34240
	104	290.32966
	131	290.32527
	282	289.71628

Table 3: Progress of the current best solution.

We implemented model (18)–(21) and Algorithm 2.1 in AMPL. In the Restoration Step, in order to compute x^k_{rest} and y^k_{rest} that satisfy (4) and (5), we solve problem (14) as in the previous section. By fixing the integer variables of the original problem to y^k, the route is determined and the problem (14) becomes the problem of finding the points $x_i \in \mathcal{R}_i$, for $i \in V$, such that the length of the route (18) is minimized. To solve problem (14), we also use Algencan version 3.0.0. To solve problem (8)–(11), we use AMPL/CPLEX version 12.1.0.

The sequence of continuous iterates generated by Algorithm 2.1 is $x^k = (x^k_1, \dots, x^k_N) \in \mathbb{R}^{nN}$, where $x^k_i \in \mathbb{R}^n$ for each $i \in V$, and the sequence of binary iterates is $y^k = (y^k_{ij})_{(i,j)\in E} \in \{0,1\}^{|E|}$. We considered three instances introduced in [21] that are available from [22]. The instances are tspn2DE5_1, tspn2DE10_1, and tspn2DE15_1, which are formed by 5, 10, and 15 ellipsoids in $\mathbb{R}^2$, respectively. Initially, we set $y^0_{i,i+1} = 1$, for each $i \in V$, $y^0_{1N} = 1$, and $y^0_{ij} = 0$ for each $(i,j) \in E$ such that $j \neq i+1$ and $(i,j) \neq (1,N)$. Also, for each $i \in V$, x^0_i is chosen uniformly random within $\mathcal{R}_i$. As in the previous experiments, we have chosen $r = 0.5$, $\beta = 1$, $\sigma_0 = 1$, and $\theta_0 = 0.1$.

Table 3 presents the results we have obtained for these three instances. It shows the evolution of the current best solution found by Algorithm 2.1. The first column presents the name of the instance. The second column shows the number of the iteration. The third column shows the value of the solution at that iteration. Figure 3 depicts the best solutions for the instances tspn2DE5_1 and tspn2DE10_1. Figure 4 illustrates the best solution found for the instance tspn2DE15_1.



tspn2DE5_1 tspn2DE10_1

Figure 3: Illustration of the solutions found for the instances tspn2DE5_1 and tspn2DE10_1.

5 Conclusions

In this paper, we consider the mixed-integer nonlinear programming problem. This general problem is very difficult to solve. Even restricting the functions to be linear or removing the integrality constraints, this problem is NP-hard [20, 30]. This problem has been tackled by different deterministic global optimization approaches [17, 33, 34], which include branch-and-bound, generalized Benders decomposition, outer approximation, the extended cutting-plane method, branch-and-cut, and reformulation-linearization/convexification, for example. However, all those global optimization strategies cannot, in general, be applied to large-scale instances like the ones that appear in practice. The method introduced in the present work can be interpreted as an heuristic that suggests which realizations of the integer variables should be explored. Moreover, this method is relatively simple to implement and can take advantage of consolidated software for solving nonlinear programming problems and mixed-integer linear/quadratic programming problems.

Acknowledgements. The authors are thankful to the anonymous referees whose comments helped to improve the quality of this work.

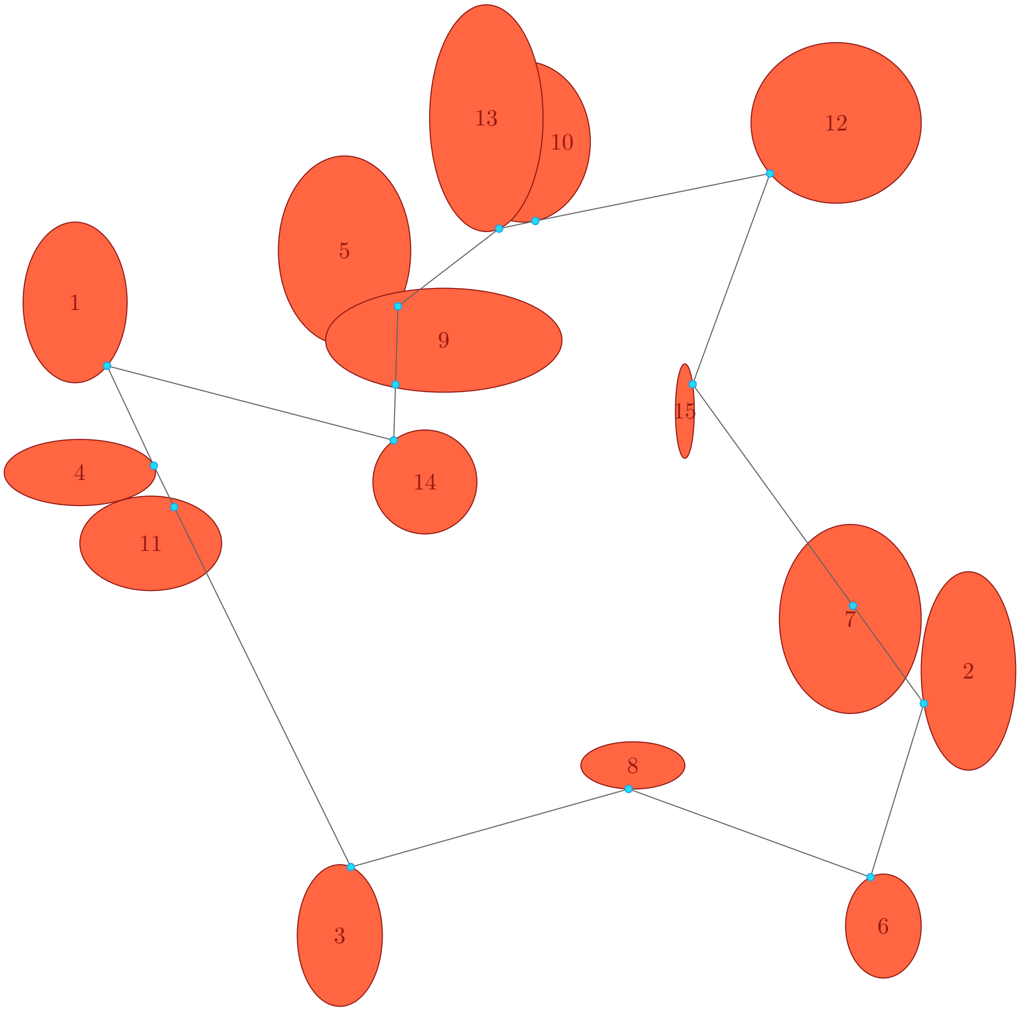


Figure 4: Illustration of the solution found for the instance tspn2DE15_1.

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization* 18, pp. 1286–1309, 2008.

[2] R. Andreani, E. G. Birgin, J. M. Martínez, and J. Yuan. Spectral projected gradient and variable metric methods for optimization with linear inequalities. *IMA Journal of Numerical Analysis* 25, pp. 221–252, 2005.

[3] R. Andreani, S. L. Castro, J. L. Chela, A. Friedlander, and S. A. Santos. An inexact-restoration method for nonlinear bilevel programming problems. *Computational Optimization and Applications* 43, pp. 307–328, 2009.

[4] M. Andretta, E. G. Birgin, and J. M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization. *Optimization* 54, pp. 305–325, 2005.

[5] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics* 55, pp. 197–218, 1994.

[6] N. Banihashemi and C. Y. Kaya. Inexact Restoration for Euler discretization of box-constrained optimal control problems. *Journal of Optimization Theory and Applications* 156, pp. 726–760, 2013.

[7] J. Barzilai and J. M. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis* 8, pp. 141–148, 1988.

[8] E. G. Birgin, L. F. Bueno, and J. M. Martínez. Assessing the reliability of general-purpose Inexact Restoration methods. *Journal of Computational and Applied Mathematics* 282, pp. 1–16, 2015.

[9] E. G. Birgin and J. M. Martínez. Local convergence of an Inexact Restoration method and numerical experiments. *Journal of Optimization Theory and Applications* 127, pp. 229–247, 2005

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

[11] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.

[12] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG - Software for convex-constrained optimization. *ACM Transactions on Mathematical Software* 27, pp. 340–349, 2001.

[13] E. G. Birgin, J. M. Martínez, and M. Raydan. Inexact Spectral Projected Gradient methods on convex sets. *IMA Journal on Numerical Analysis* 23, pp. 539–559, 2003.

[14] L. F. Bueno, A. Friedlander, J. M. Martínez, and F. N. C. Sobral. Inexact Restoration method for derivative-free optimization with smooth constraints. *SIAM Journal on Optimization* 23, pp. 1189–1231, 2013.

[15] A. Fischer and A. Friedlander. A new line search Inexact Restoration approach for nonlinear programming. *Computational Optimization and Applications* 46, pp. 333–346, 2010.

[16] R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximations. *Mathematical Programming* 66, pp. 327–349, 1994

[17] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications.* Oxford University Press, London, 1995.

[18] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL – A Modeling Language for Mathematical Programming.* Duxbury Press, 2nd edition, 2002.

[19] A. Friedlander, J. M. Martínez, and M. Raydan. A new method for large-scale box constrained convex quadratic minimization problems. *Optimization Methods and Software* 5, pp. 57–74, 1995.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, NY, USA, 1979.

[21] I. Gentilini, F. Margot and K. Shimada. The Travelling Salesman Problem with Neighbourhoods: MINLP Solution. *Optimization Methods Software*, 28, pp. 364–378, 2013.

[22] I. Gentilini, F. Margot, and K. Shimada. STSPN Instances. URL `http://wpweb2.tepper.cmu.edu/fmargot/ampl.html`.

[23] M. A. Gomes-Ruggiero, J. M. Martínez, and S. A. Santos. Spectral projected gradient method with Inexact Restoration for minimization with nonconvex constraints. *SIAM Journal on Scientific Computing* 31, pp. 1628–1652, 2009.

[24] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization Enginneering* 3, pp. 227–252, 2002.

[25] IBM, IBM ILOG CPLEX Optimizer, High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming, `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`, 2011.

[26] C. Y. Kaya and J. M. Martínez. Euler discretization and Inexact Restoration for optimal control. *Journal of Optimization Theory and Applications* 134, pp. 191–206, 2007.

[27] N. Krejić and J. M. Martínez. Inexact Restoration approach for minimization with inexact evaluation of the objective function. *Mathematics of Computation* 85, pp. 1775–1791, 2016.

[28] J. M. Martínez. Inexact Restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. *Journal of Optimization Theory and Applications* 111, pp. 39–58, 2001.

[29] J. M. Martínez and E. A. Pilotta. Inexact restoration algorithms for constrained optimization. *Journal of Optimization Theory and Applications* 104, pp. 135–163, 2000.

[30] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming* 39, pp. 117–129, 1987.

[31] M. Raydan. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis* 13, pp. 321–326, 1993.

[32] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization* 7, pp. 26–33, 1997.

[33] H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Springer US, 1999.

[34] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Springer US, 2002.