



# Using regularization and second order information in outer approximation for convex MINLP

Jan Kronqvist<sup>1</sup> · David E. Bernal<sup>2</sup> · Ignacio E. Grossmann<sup>2</sup>

Received: 14 December 2017 / Accepted: 5 December 2018 / Published online: 18 December 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2018

## Abstract

In this paper, we present two new methods for solving convex mixed-integer nonlinear programming problems based on the outer approximation method. The first method is inspired by the level method and uses a regularization technique to reduce the step size when choosing new integer combinations. The second method combines ideas from both the level method and the sequential quadratic programming technique and uses a second order approximation of the Lagrangean when choosing the new integer combinations. The main idea behind the methods is to choose the integer combination more carefully at each iteration, in order to obtain the optimal solution in fewer iterations compared to the original outer approximation method. We prove rigorously that both methods will find and verify the optimal solution in a finite number of iterations. Furthermore, we present a numerical comparison of the methods based on 109 test problems to illustrate their advantages.

**Mathematics Subject Classification** 65K05 · 90C11 · 90C30 · 90C55

## 1 Introduction

Mixed-integer nonlinear programming (MINLP) is a class of optimization problems containing both integer and continuous variables as well as nonlinear functions. The

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s10107-018-1356-3>) contains supplementary material, which is available to authorized users.

---

✉ Ignacio E. Grossmann  
grossmann@cmu.edu

Jan Kronqvist  
jan.kronqvist@abo.fi

David E. Bernal  
debernal@andrew.cmu.edu

<sup>1</sup> Process Design and Systems Engineering, Åbo Akademi University, Turku, Finland

<sup>2</sup> Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

integer variables make it possible to incorporate logic relations and discrete quantities in the mathematical model. Together with linear and nonlinear constraints, MINLP becomes a powerful framework for modeling real-world optimization problems, and thus, there is a vast number of applications in areas such as engineering, computational chemistry, and finance [3,14]. MINLP problems are by definition non-convex; however, they are still commonly classified as either convex or non-convex. An MINLP problem is considered as convex if an integer relaxation results in a convex nonlinear programming (NLP) problem [25]. Convexity is a desirable property since it enables the direct use of several decomposition techniques for solving the problem. Such decomposition techniques are, e.g., outer approximation (OA) [10], extended cutting plane (ECP) [34], extended supporting hyperplane (ESH) [23], generalized Benders decomposition (GBD) [16], and branch and bound (BB) techniques [8]. For reviews of MINLP methods and applications see [2,4,25,30]. Even if there are several methods available for solving convex MINLP problems, it is still a challenging type of optimization problems as shown in the solver benchmark in [23].

Methods such as OA, ECP, ESH, and GBD all generate an iteratively improving linear approximation of the MINLP problem, where the nonlinear functions are underestimated by first-order Taylor series expansions. The linear approximation is a mixed-integer linear programming (MILP) problem and is often referred to as the MILP-master problem. All these methods iteratively choose the integer trial solutions as the minimizer of the MILP-master problem. Choosing the iterative solutions as the minimizer of a linear approximation is similar to the approach used in Kelley's method [21], which is an algorithm intended for convex NLP problems. It is known that Kelley's method is not efficient at handling nonlinearities and it has a poor complexity bound, see, e.g., [27]. Kelley's method is sometimes even referred to as unstable since the iterative solutions tend to make large jumps in the search space [11]. Since methods such as ECP, ESH, GBD, and OA choose the iterative integer solutions in the same manner as Kelley's method, they could also suffer from the same instability. Several techniques to reduce the instability of Kelley's method have successfully been used for NLP problems, e.g., regularization to reduce the step size or the concept of a trust region [1].

Due to the non-convex nature of MINLP problems, it is not trivial to use regularization of the step size or a trust region when solving such problems, since the integer requirements may cause solutions to be far apart in the search space. However, recently there has been interest in the idea of using regularization for solving convex MINLP problems, e.g., using quadratic stabilization with Benders decomposition was proposed in [37] and using regularization combined with a cutting plane method was presented in [12].

Here we present an approach for introducing stabilization in the subproblems for choosing the integer combination in OA. The stabilization technique is inspired by the regularization used in the level method for NLP, see [22,26], and therefore, the method is referred to as level-based outer approximation (L-OA). By modifying the L-OA method it is possible to include second order information in the subproblems of choosing the integer combination, and we refer to this method as quadratic outer approximation (Q-OA). In Q-OA we use a second order Taylor series expansion for the Lagrangean function as the objective in the subproblems for finding a new integer combination. A similar quadratic approach was presented in [13]. However, the

level constraint used in the level method provides a more robust way of enforcing an improvement and avoiding cycling. Furthermore, the level constraint forces the solutions to be chosen as an interpolation between the minimizer of the Lagrangean approximation and the minimizer of the linear approximation in the MILP-master problem. The proposed methods are motivated by the strong convergence properties of the level method compared to Kelley's, and recent advances in software for solving MILP and mixed-integer quadratic programming (MIQP) problems.

The proposed methods are intended to accelerate the convergence of OA by choosing the integer combinations more carefully, using either a regularization technique or second-order information. Due to the regularization and the use of second-order derivatives, the proposed methods should be better suited for handling nonlinearities compared to OA. However, each iteration in L-OA and Q-OA will also be more complex than an iteration in OA. For MINLP problems with only a few nonlinear terms, there might not be significant improvements by the proposed methods. The methods are, thus, mainly intended for problems with moderate to high degree of nonlinearity. We begin with a brief review of OA in Sect. 2, and from there we continue by presenting the basics of L-OA and Q-OA in Sects. 3 and 4. In Sect. 5, it is proven that the convergence properties of OA still hold with the modifications in the proposed methods. Finally, in Sect. 6 we present a numerical comparison of Q-OA, L-OA, and OA, based on test problems from the problem library MINLib2 [15].

## 2 Background

The MINLP problems considered here can be written as follows,

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & f(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} \quad & g_j(\mathbf{x}, \mathbf{y}) \leq 0 \quad \forall j = 1, \dots, l, \\ & \mathbf{Ax} + \mathbf{By} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Z}^m. \end{aligned} \tag{MINLP}$$

In order to guarantee global convergence, we need to assume some properties of the nonlinear functions. Throughout this paper we rely on the following assumptions:

**Assumption 1** The nonlinear functions  $f, g_1, \dots, g_l : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  are convex and continuously differentiable.

**Assumption 2** The linear constraints define a nonempty compact set.

**Assumption 3** For each feasible integer combination  $\mathbf{y}$ , an integer combination such that there exist  $\mathbf{x}$  variables for which the problem is feasible, a constraint qualification holds, e.g., Slater's condition [29].

These are the typical assumptions needed for rigorously proving convergence of OA, see [10, 13]. OA can be generalized to be applicable to non-differentiable problems, see, e.g., [33], although such problems are not considered here.

We begin by briefly presenting the main steps of the outer approximation method. As previously mentioned, the method uses a linear approximation of the MINLP

problem to obtain trial solutions for the integer variables. Once an integer combination is obtained, the corresponding continuous variables can be determined by solving a continuous optimization problem. The previously obtained trial solutions  $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=0}^k$  are used to construct the linear approximation of the MINLP problem. At iteration  $k$ , the next integer combination  $\mathbf{y}^{k+1}$  is obtained by solving the following MILP subproblem

$$\begin{aligned}
 & \min_{\mathbf{x}, \mathbf{y}, \mu} \mu \\
 \text{s.t.} \quad & f(\mathbf{x}^i, \mathbf{y}^i) + \nabla f(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq \mu \quad \forall i = 1, \dots, k, \\
 & g_j(\mathbf{x}^i, \mathbf{y}^i) + \nabla g_j(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq 0 \quad \forall i = 1, \dots, k, \forall j \in \mathcal{I}_i, \\
 & \mathbf{Ax} + \mathbf{By} \leq \mathbf{b}, \\
 & \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Z}^m, \mu \in \mathbb{R}.
 \end{aligned}
 \tag{OA-master}$$

Here  $\mathcal{I}_i$  are index sets containing the indexes of the nonlinear constraints active at the trial solution  $(\mathbf{x}^i, \mathbf{y}^i)$  [13]. Due to convexity, we know that the feasible set is overestimated and that the objective will be underestimated, see, e.g., [10] and Lemma 1 in Sect. 5. The optimum of problem (OA-master), thus, provides a valid lower bound to the MINLP problem, which is referred to as  $LB^{k+1}$ . Once the new integer combination  $\mathbf{y}^{k+1}$  is obtained, the corresponding  $\mathbf{x}$  variables can be obtained by solving the following convex NLP subproblem,

$$\begin{aligned}
 & \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^{k+1}) \\
 \text{s.t.} \quad & g_j(\mathbf{x}, \mathbf{y}^{k+1}) \leq 0 \quad \forall j = 1, \dots, l, \\
 & \mathbf{Ax} + \mathbf{By}^{k+1} \leq \mathbf{b}, \\
 & \mathbf{x} \in \mathbb{R}^n.
 \end{aligned}
 \tag{NLP-I}$$

If problem (NLP-I) is feasible and solved to optimality, we obtain  $\mathbf{x}^{k+1}$  and furthermore, the optimum provides a valid upper bound  $UB^{k+1}$  to the MINLP problem. Otherwise, if the NLP problem is infeasible, we need a different approach to obtain the  $\mathbf{x}$  variables and this can be done, for example, by solving a feasibility problem. The feasibility problem minimizes the constraint violation with the current choice of  $\mathbf{y}$  variables, e.g., using the  $\ell_\infty$  norm, and it can be defined as,

$$\begin{aligned}
 & \min_{\mathbf{x}, r} r \\
 \text{s.t.} \quad & g_j(\mathbf{x}, \mathbf{y}^{k+1}) \leq r \quad \forall j = 1, \dots, l, \\
 & \mathbf{Ax} + \mathbf{By}^{k+1} \leq \mathbf{b}, \\
 & \mathbf{x} \in \mathbb{R}^n, r \in \mathbb{R}_+.
 \end{aligned}
 \tag{NLP-f}$$

By solving problem (NLP-f) the continuous variables  $\mathbf{x}^{k+1}$  are obtained. However, in this case,  $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$  is not a feasible solution, and thus, no upper bound is obtained at this iteration. The feasibility problem always satisfies Slater’s condition and due to the convexity assumption, we know that the feasibility problem is always feasible and tractable.

In case the difference between the upper and lower bound is not within the desired tolerance, we improve the linear approximation by adding new linearizations to problem (OA-master). These linearizations are often referred to as cutting planes or supporting hyperplanes, and they are given by,

$$\begin{aligned} f(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}) + \nabla f(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^{k+1} \\ \mathbf{y} - \mathbf{y}^{k+1} \end{bmatrix} &\leq \mu, \\ g_j(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}) + \nabla g_j(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^{k+1} \\ \mathbf{y} - \mathbf{y}^{k+1} \end{bmatrix} &\leq 0 \quad \forall j \in \mathcal{I}_{k+1}. \end{aligned} \quad (1)$$

Due to convexity, the cuts will not exclude any feasible solution from the search space [6]. Adding these cuts to the MILP subproblem ensures that the integer combination  $\mathbf{y}^{k+1}$  will not be obtained in a consecutive iteration unless it is the optimal integer solution. Convergence can be ensured since each iteration will either result in a new integer combination or verify optimality. For more details of OA see [10, 13, 31]. The basic steps of OA are summarized as a pseudo-code in Algorithm 1.

---

**Algorithm 1** An algorithm summarizing the basic steps of the outer approximation method

---

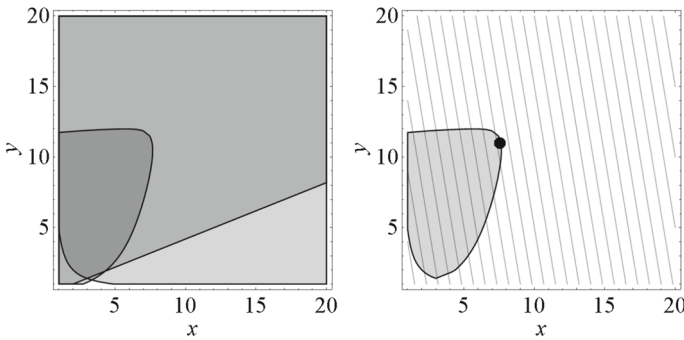
Define accepted optimality gap  $\epsilon \geq 0$ .

1. Initialization.
    - 1.1 Obtain a relaxed solution  $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$  by solving an integer relaxation of the MINLP problem.
    - 1.2 Generate cuts at  $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$  according to (1) and construct problems (OA-master).
    - 1.3 Set iteration counter  $k = 1$ ,  $UB^0 = \text{inf}$  and  $LB^0 = -\text{inf}$ .
  2. Repeat until  $UB^{k-1} - LB^{k-1} \leq \epsilon$ .
    - 2.1 Solve problem (OA-master) to obtain  $\mathbf{y}^k$  and  $LB^k$
    - 2.4 Solve problem (NLP-I) with integer variables fixed as  $\mathbf{y}^k$  to obtain  $\mathbf{x}^k$ .
      - 2.4.1 If problem (NLP-I) is infeasible, obtain  $\mathbf{x}^k$  by solving feasibility problem (NLP-f) and set  $UB^k = UB^{k-1}$ .
    - 2.5 Generate cuts at  $\mathbf{x}^k, \mathbf{y}^k$  according to (1) and add these to problems (OA-master).
    - 2.6 If  $\mathbf{x}^k, \mathbf{y}^k$  is feasible, set  $UB^k = \min\{f(\mathbf{x}^k, \mathbf{y}^k), UB^{k-1}\}$ .
    - 2.7 Increase iteration counter,  $k = k + 1$
  - 3 Return the best found solution.
- 

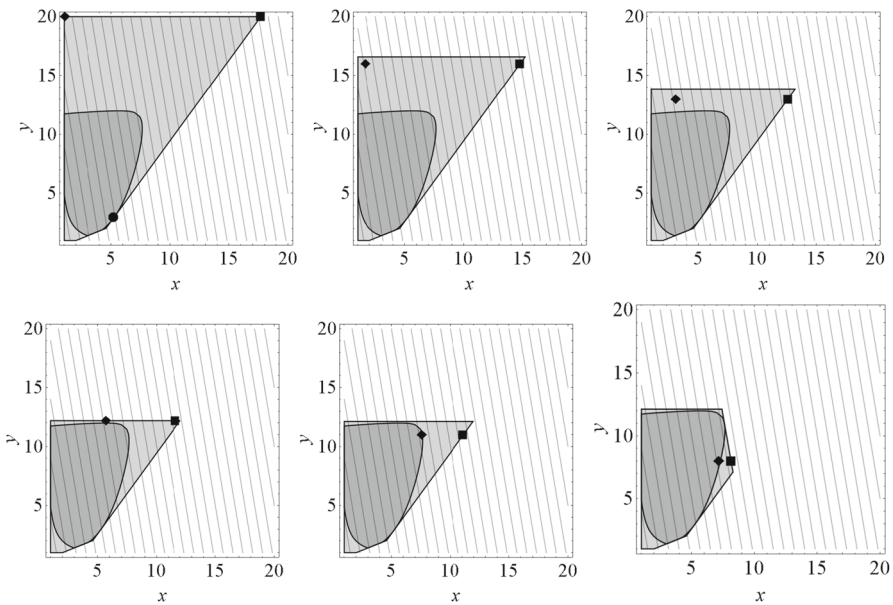
Here we have not considered the integer cuts used in [10], since these are not needed for convex problems. To get a better understanding of OA and to highlight the differences compared to the other methods, consider the following simple example

$$\begin{aligned} &\text{minimize } -6x - y \\ &\text{s.t.} \quad 0.3(x - 8)^2 + 0.04(y - 6)^4 + 0.1e^{2x}y^{-4} \leq 56 \\ &\quad 1/x + 1/y - x^{0.5}y^{0.5} \leq -4 \\ &\quad 2x - 5y \leq -1 \\ &\quad 1 \leq x \leq 20, \quad 1 \leq y \leq 20, \quad x \in \mathbb{R}, \quad y \in \mathbb{Z}. \end{aligned} \quad (\text{Ex } 1)$$

The basic features of problem (Ex 1) are illustrated in Fig. 1, showing the constraints, objective, and the optimal solution.



**Fig. 1** The figure to the left shows the feasible regions of the constraints in problem (Ex 1). The second figure shows the integer relaxed feasible region, contours of the objective and the optimal solution



**Fig. 2** The figures show the feasible region defined by the nonlinear constraints in dark gray, and the light gray areas show the outer approximation obtained by the generated cuts. The squared dots represent the solutions obtained from the MILP subproblem and diamond shaped dots represent the solutions obtained by one of the NLP subproblems. The dot in the first figure shows the starting point  $(x^0, y^0)$

Later, we use the same example to illustrate the differences between the original OA and the proposed methods. To make the results comparable, we will use the starting point,  $x^0 = 5.29, y^0 = 3$  with all the methods. Instead of solving the relaxed problem in the initialization step in Algorithm 1, we simply use  $(x^0, y^0)$  as a starting point. OA required 7 iterations to solve this problem, of which the first six iterations are shown in Fig. 2. For this specific problem, the first four iterations all result in infeasible solutions where one of the nonlinear constraints are violated. The optimal solution is obtained in iteration five, but verifying optimality requires two additional iterations.

Next, we will show how ideas from the level method can be combined with OA to obtain a stabilized approach for choosing new integer combinations.

### 3 Level-based OA

The level method was originally presented in [26], as a method for solving non-smooth NLP problems. Like OA, the level method also constructs a linear approximation of the original optimization problem. However, the trial solutions are not chosen as the minimizer of the linear approximation. Instead, the trial solutions are obtained by projecting the current solution onto a specific level set of the linearly approximated objective function. For more details see [22,27]. Here we will use a similar approach combined with OA, which we show is equivalent to adding specific trust regions to the problems (OA-master) in the original OA.

Here we assume that a feasible solution to the MINLP problem  $\bar{\mathbf{x}}, \bar{\mathbf{y}}$  is known. Such a solution can for example be obtained by first performing some original OA iterations or by using a specific procedure such as the feasibility pump [5]. An upper bound to the MINLP problem is, thus, given by  $f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$  and cuts at  $\bar{\mathbf{x}}, \bar{\mathbf{y}}$  can be generated according to (1) to form problem (OA-master). A valid lower bound  $LB^1$  can be obtained by solving the linear subproblem (OA-master), and thus, the bounds for the optimal objective  $f^*$  are given by  $LB^1 \leq f^* \leq f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ .

From the bounds of the optimal objective we can in each iteration  $k$  estimate a value of the optimal objective according to,

$$\hat{f}_k^* = (1 - \alpha)f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) + \alpha LB^k, \quad (2)$$

where  $\alpha \in (0, 1]$  and  $\bar{\mathbf{x}}, \bar{\mathbf{y}}$  is chosen as the best found feasible solution, similarly as in the level method. The lower bound  $LB^k$  is obtained as in the original OA, by solving problem (OA-master). In Eq. (2)  $\alpha$  is a parameter which represents how much the linear approximation of the MINLP problem is trusted. Setting  $\alpha$  close to one results in an estimated optimum  $\hat{f}_k^*$  close to the lower bound, while setting it close to zero results in an estimated optimum close to the best incumbent solution. The next integer solution  $\mathbf{y}^{k+1}$  can now be obtained by projecting  $\bar{\mathbf{x}}, \bar{\mathbf{y}}$  onto the  $\hat{f}_k^*$  level set of the linearly approximated objective function. The projection is performed by solving the following MIQP problem,

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{y}, \mu} \left\| \begin{array}{c} \mathbf{x} - \bar{\mathbf{x}} \\ \mathbf{y} - \bar{\mathbf{y}} \end{array} \right\|^2 \\ & \text{s.t. } \mu \leq \hat{f}_k^* \\ & \quad f(\mathbf{x}^i, \mathbf{y}^i) + \nabla f(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq \mu \quad \forall i = 1, \dots, k, \\ & \quad g_j(\mathbf{x}^i, \mathbf{y}^i) + \nabla g_j(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq 0 \quad \forall i = 1, \dots, k, \forall j \in \mathcal{I}_i, \\ & \quad \mathbf{Ax} + \mathbf{By} \leq \mathbf{b}, \\ & \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Z}^m, \mu \in \mathbb{R}, \end{aligned}$$

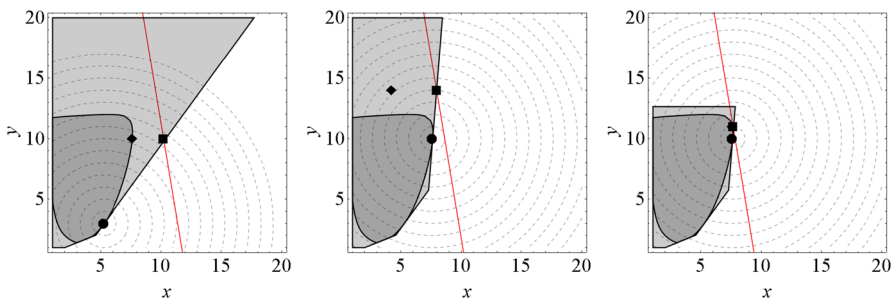
(MIQP-Proj)

where  $\|\cdot\|$  is the Euclidean norm. The MIQP problem should contain all the supporting hyperplanes and cutting planes present in problem (OA-master), which was solved to obtain the lower bound. The objective function of problem (MIQP-Proj) introduces a regularization to each iteration, by penalizing the change from the best known solution (the step size). The next integer solution  $\mathbf{y}^{k+1}$  is thereby chosen as a point as close as possible to the best known feasible solution, which reduces the linearly approximated objective to at most  $\hat{f}_k^*$ . Since  $\hat{f}_k^*$  is calculated according to (2) there always exists a solution to the MIQP problem, e.g., the minimizer of problem (OA-master) will satisfy all the constraints. Once the new integer combination is obtained, the corresponding continuous variables can be determined using the same technique as described in the previous section. We summarize the level based outer approximation as a pseudo-code in Algorithm 2.

The regularization will not only reduce the step size between the iterative solutions, but it will also try to keep the trial solutions close to the best known solution and simultaneously close to the feasible set. This gives an advantage over the original OA, especially, in early iterations where the linear MILP-master problems might only provide a poor approximation which can result in trial solutions far from the feasible set.

The main difference of L-OA compared to OA is the two-step procedure for obtaining the new integer combination, which involves both the solution of an MILP and an MIQP subproblem. This increases the complexity of each iteration. However, as we will prove later the MIQP need not to be solved to optimality. Basically, any feasible solution to the MIQP will be sufficient for ensuring convergence. The computational aspects are described in more detail in Sect. 6 and convergence of both L-OA and Q-OA is proved in Sect. 5.

To obtain a geometrical understanding of how L-OA differs from the original OA, we again consider problem (Ex 1). Here we use the same starting point as before and we set the level parameter as  $\alpha = 0.4$ . To solve the problem with these parameters L-OA requires four iterations. The three first iterations are shown in Fig. 3. In the fourth iteration, we are able to verify optimality directly after solving the MILP subproblem since we obtain  $LB^4 = f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ .



**Fig. 3** The figure illustrates the first three iterations needed to solve problem (Ex 1) with the L-OA method. The dashed circles represent the contours of the objective function in the MIQP subproblems and the red line shows the level constraint given by  $\mu \leq \hat{f}_k^*$ . The circular dots represent the best-found solution so far, the squared dots represent the solutions obtained from the MIQP subproblem and diamond shaped dots represent the solutions obtained by one of the NLP subproblems



---

**Algorithm 2** An algorithm summarizing the basic steps of level-based outer approximation (L-OA)

---

Define accepted optimality gap  $\epsilon \geq 0$  and choose the parameter  $\alpha \in (0, 1]$ .

1. Initialization.
    - 1.1 Obtain a feasible solution  $\bar{x}, \bar{y}$ , either by OA or by any other technique.
    - 1.2 Generate cuts at  $\bar{x}, \bar{y}$  according to (1) and construct problems (OA-master) and (MIQP-Proj).
    - 1.3 Set iteration counter  $k = 1$ , and  $LB^0 = -\inf$ .
  2. Repeat until  $f(\bar{x}, \bar{y}) - LB^{k-1} \leq \epsilon$ .
    - 2.1 Solve problem (OA-master) to obtain  $LB^k$
    - 2.2 Calculate the estimated optimal value  $\hat{f}_k^*$  according to (2).
    - 2.3 Solve problem (MIQP-Proj) to obtain  $y^k$
    - 2.4 Solve problem (NLP-I) with integer variables fixed as  $y^k$  to obtain  $x^k$ .
      - 2.4.1 If problem (NLP-I) is infeasible, obtain  $x^k$  by solving feasibility problem (NLP-f).
    - 2.5 Generate cuts at  $x^k, y^k$  according to (1) and add these to problems (OA-master) and (MIQP-Proj).
    - 2.6 If  $x^k, y^k$  is feasible and  $f(x^k, y^k) \leq f(\bar{x}, \bar{y})$ , set  $\bar{x}, \bar{y} = x^k, y^k$ .
    - 2.7 Increase iteration counter,  $k = k + 1$
  - 3 Return  $\bar{x}, \bar{y}$  as the optimal solution.
- 

As mentioned earlier, L-OA will find similar integer solutions as adding specific trust regions to the MILP subproblems in the original OA. This property is further described in Theorem 1.

**Theorem 1** *The procedure of solving problems (OA-master) and (MIQP-Proj) will result in a solution equivalent to adding the trust region constraint*

$$\left\| \begin{array}{l} x - \bar{x} \\ y - \bar{y} \end{array} \right\|^2 \leq r_k, \quad (3)$$

to problem (OA-master) in the original OA, where  $r_k$  is chosen as the optimum of problem (MIQP-Proj).

**Proof** First, assume that there exists a unique solution to problem (MIQP-Proj), and denote the minimizer as  $x^{\text{MIQP}}, y^{\text{MIQP}}, \mu^{\text{MIQP}}$ . As stated the radius of the trust region constraint is chosen as

$$r_k = \left\| \begin{array}{l} x^{\text{MIQP}} - \bar{x} \\ y^{\text{MIQP}} - \bar{y} \end{array} \right\|^2. \quad (4)$$

Adding the trust region constraint given by Eq. (3) with radius  $r_k$  to problem (OA-master) gives the solution  $x^{\text{MILP}}, y^{\text{MILP}}, \mu^{\text{MILP}}$ . Now, assume this solution is not the same as the MIQP solution. Since the MIQP solution is assumed to be unique and not equal to the MILP solution, it follows that,

$$r_k > \left\| \begin{array}{l} x^{\text{MILP}} - \bar{x} \\ y^{\text{MILP}} - \bar{y} \end{array} \right\|^2. \quad (5)$$

Furthermore, since (OA-master) minimizes  $\mu$  we get  $\mu^{\text{MILP}} \leq \mu^{\text{MIQP}} \leq \hat{f}_k^*$ . This leads to a contradiction since  $\mathbf{x}^{\text{MILP}}, \mathbf{y}^{\text{MILP}}, \mu^{\text{MILP}}$  would then define a feasible solution to problem (MIQP-Proj) with an objective strictly lower than the solution obtained by solving the minimization problem.

Next, we consider the case where there is not a unique optimal solution to problem (MIQP-Proj), but multiple optimal solutions. As before, we assume that  $\mathbf{x}^{\text{MILP}}, \mathbf{y}^{\text{MILP}}, \mu^{\text{MILP}}$  is not an optimal solution to problem (MIQP-Proj). However, Eq. (5) must still hold with strict inequality, since the MILP solution satisfies the trust region constraint given by Eq. (3) and it is not an optimal solution to problem (MIQP-Proj). This leads to the same contradiction as in the case of a unique solution, and therefore, the MILP solution must be an optimal solution to problem (MIQP-Proj).  $\square$

Note that there are no practical implications that follow from Theorem 1 because the radius of the trust region resulting in similar solutions cannot be determined in advance. However, the theorem proves that the procedure used in L-OA can be viewed as a technique of using a trust region with OA. Next, we show that it is possible to use a similar approach as L-OA to incorporate second order information in the task of obtaining the integer combinations.

### 4 Quadratic outer approximation

In order to obtain better integer solutions, it would be desirable to use information regarding the curvature of the constraints and objective in the task of choosing the integer combinations. We propose a technique where second-order information is incorporated by minimizing a second order Taylor series expansion of the Lagrangean function, which was also suggested in [13]. By using the Lagrangean it is possible to include curvature of both the constraints and objective while keeping the constraints of the subproblems linear.

Here we define the Lagrangean function  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \rightarrow \mathbb{R}$  as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \lambda) = f(\mathbf{x}, \mathbf{y}) + \sum_{j=1}^l \lambda_j g_j(\mathbf{x}, \mathbf{y}), \tag{6}$$

where  $\lambda_j \geq 0$  is the Lagrange multiplier of the  $j$ -th nonlinear constraint. We do not include the linear constraints in the Lagrangean, since these are handled directly in the subproblems. The Lagrangean is frequently used in NLP techniques and has the following important properties.

**Property 1** *If all nonlinear functions  $f, g_1, \dots, g_j$  in problem (MINLP) are convex, then for nonnegative multipliers the Lagrangean defined in Eq. (6) will be a convex function in the  $\mathbf{x}, \mathbf{y}$  variables, see, e.g., [6,36].*

**Property 2** *Strong duality holds for convex optimization problems that satisfy Slater’s condition; i.e., there exists valid multipliers such that the minimum of the Lagrangean is equal to the minimum of the original problem [6].*

Since the MINLP problems are non-convex by nature, we cannot expect strong duality to hold. However, the first property is important since it will ensure that the subproblem we use for finding the integer combinations will be tractable. We do not want to directly minimize the Lagrangean, because that problem is basically as difficult as the original problem. Therefore, we will use a second order approximation of the Lagrangean, which is given by

$$\mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\lambda}) + \nabla_{\mathbf{x}, \mathbf{y}} \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\lambda})^T \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix}^T \nabla_{\mathbf{x}, \mathbf{y}}^2 \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\lambda}) \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix}, \quad (7)$$

where  $\nabla_{\mathbf{x}, \mathbf{y}} \mathcal{L}$  is the gradient of the Lagrangean with respect to  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\nabla_{\mathbf{x}, \mathbf{y}}^2$  denotes the Hessian matrix. To make the notation more compact we have introduced the  $\Delta$ -variables that are given by  $\Delta \mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$  and  $\Delta \mathbf{y} = \mathbf{y} - \bar{\mathbf{y}}$ . Due to Property 1, we know that the Hessian  $\nabla_{\mathbf{x}, \mathbf{y}}^2$  will be positive semidefinite for all  $\lambda \geq \mathbf{0}$ . For small changes in the  $\Delta$ -variables Eq. (7) should give a good approximation, although the approximation does not under- or overestimate the real Lagrangean function.

The natural approach of using the quadratic approximation in OA would be to replace the linear objective of the MILP-master problem with the quadratic function given by Eq. (7). However, this approach will not guarantee convergence on its own, because, unlike the original OA the quadratic master problem will not always result in new integer combinations. Since the second order approximation does not necessarily underestimate the Lagrangean, it is possible that the approximation point  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{y}}$  is the optimum of the approximation even if it is not the optimal solution to the original problem, and thus, the approach could stagnate at non-optimal solutions. To avoid this, the method presented in [13] uses an  $\epsilon$  improvement strategy, where the next solution must reduce the linearly approximated objective by a small  $\epsilon$ -value. The  $\epsilon$  improvement is enforced by the following constraints

$$\begin{aligned} \mu &\leq f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - \epsilon \\ f(\mathbf{x}^i, \mathbf{y}^i) + \nabla f(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} &\leq \mu \quad \forall i = 1, \dots, k, \end{aligned} \quad (8)$$

where  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{y}}$  is the best found solution. With this approach  $\epsilon$  must be chosen smaller than the desired optimality gap. Thus, it will only result in a small reduction requirement. Therefore, the quadratic outer approximation method in [13] will rely heavily on the second order approximation of the Lagrangean. In case the approximation point  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{y}}$  with the corresponding multipliers  $\bar{\lambda}$  is not the optimal solution to the MINLP, then the Lagrangean might not give a good approximation of the original problem and this might cause slow convergence. Due to the discrete nature of MINLP problems, it is possible that only the optimal integer combination with the corresponding continuous variables will result in the optimal set of active constraints and nonzero multipliers.

Here, we use a different approach, which combines information from both the linear approximation with the quadratic approximation of the Lagrangean, to make sure the proposed method does not stagnate at non-optimal solutions. By using the

same approach as in L-OA, an estimate of the optimal objective  $\hat{f}_k^*$  can be calculated according to Eq. (2). The estimated optimum can further be used to construct the following reduction constraint,

$$\begin{aligned} \mu &\leq \hat{f}_k^* \\ f(\mathbf{x}^i, \mathbf{y}^i) + \nabla f(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} &\leq \mu \quad \forall i = 1, \dots, k. \end{aligned} \tag{9}$$

As long as  $\hat{f}_k^*$  is calculated using the same technique as in L-OA, there will always exist a solution that satisfies the reduction constraints in Eq. (9). Furthermore, since  $\hat{f}_k^*$  is chosen as an interpolation between the upper and lower bound it will usually result in a stricter reduction constraint. We will construct the master problem by minimizing the quadratic approximation of the Lagrangean with the reduction constraint given by Eq. (9), the accumulated cuts given by Eq. (1) and all linear constraints from the MINLP problem. The new integer combination  $y^{k+1}$  is, thus, obtained by solving the following MIQP problem,

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}, \mu} \quad & \nabla_{\mathbf{x}, \mathbf{y}} \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\lambda})^T \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix}^T \nabla_{\mathbf{x}, \mathbf{y}}^2 \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\lambda}) \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} \\ \text{s.t.} \quad & \mu \leq \hat{f}_k^* \\ & f(\mathbf{x}^i, \mathbf{y}^i) + \nabla f(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq \mu \quad \forall i = 1, \dots, k \\ & g_j(\mathbf{x}^i, \mathbf{y}^i) + \nabla g_j(\mathbf{x}^i, \mathbf{y}^i)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^i \\ \mathbf{y} - \mathbf{y}^i \end{bmatrix} \leq 0 \quad \forall i = 1, \dots, k, \forall j \in \mathcal{I}_i, \\ & \mathbf{Ax} + \mathbf{By} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{Z}^m, \mu \in \mathbb{R}, \end{aligned} \tag{QOA-master}$$

where  $\Delta \mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$  and  $\Delta \mathbf{y} = \mathbf{y} - \bar{\mathbf{y}}$ . As in L-OA  $\bar{\mathbf{x}}, \bar{\mathbf{y}}$  is chosen as the best found feasible solution and  $\bar{\lambda}$  are the corresponding Lagrange multipliers obtained by solving problem (NLP-1). The NLP subproblem with fixed integer variables will provide both the  $\mathbf{x}$  variables and the multipliers  $\lambda$ . If the NLP subproblem is infeasible we solve the problem (NLP-f), from which we obtain the corresponding multipliers. As mentioned before  $\nabla_{\mathbf{x}, \mathbf{y}}^2 \mathcal{L}$  is positive semidefinite due to the convexity of the nonlinear functions; therefore, the MIQP problem can be solved efficiently with software such as Gurobi [19] or Cplex [20].

Once the next integer solution has been obtained, the continuous variables are determined as in OA or L-OA, and more cuts are generated according to Eq. (1). The lower bound is updated in each iteration as in L-OA by solving problem (OA-master). The quadratic outer approximation method is summarized as a pseudocode in Algorithm 3.

As in L-OA, each iteration includes both an MILP and an MIQP subproblem. We will show later that it is sufficient to merely obtain a feasible solution to the MIQP,

---

**Algorithm 3** An algorithm summarizing the basic steps of the quadratic outer approximation (Q-OA) method

---

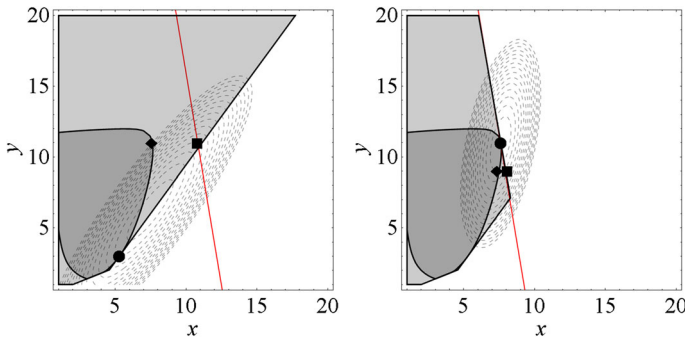
Define accepted optimality gap  $\epsilon \geq 0$  and choose the parameter  $\alpha \in ]0, 1]$ .

1. Initialization.
    - 1.1 Obtain a feasible solution  $\bar{x}, \bar{y}$  and the multipliers  $\bar{\lambda}$ , either by OA or by any other technique.
    - 1.2 Generate cuts at  $\bar{x}, \bar{y}$  according to (1) and construct problems (OA-master) and (QOA-master).
    - 1.3 Set iteration counter  $k = 1$ , and  $LB^0 = -\inf$ .
  2. Repeat until  $f(\bar{x}, \bar{y}) - LB^{k-1} \leq \epsilon$ .
    - 2.1 Solve problem (OA-master) to obtain  $LB^k$
    - 2.2 Calculate the estimated optimal value  $\hat{f}_k^*$  according to (2).
    - 2.3 Solve problem (QOA-master) to obtain  $y^k$
    - 2.4 Solve problem (NLP-I) with integer variables fixed as  $y^k$  to obtain  $x^k$  and  $\lambda^k$ .
      - 2.4.1 If problem (NLP-I) is infeasible, obtain  $x^k$  by solving feasibility problem (NLP-f).
    - 2.5 Generate cuts at  $x^k, y^k$  according to (1) and add these to problems (OA-master) and (QOA-master).
    - 2.6 If  $x^k, y^k$  is feasible and  $f(x^k, y^k) \leq f(\bar{x}, \bar{y})$ , set  $\bar{x}, \bar{y}, \bar{\lambda} = x^k, y^k, \lambda^k$ .
    - 2.7 Increase iteration counter,  $k = k + 1$
  - 3 Return  $\bar{x}, \bar{y}$  as the optimal solution.
- 

which can reduce the computational complexity of both the L-OA and Q-OA method. Section 5 proves the method's convergence to the optimal solution in a finite number of iterations, and Sect. 6 discusses the computational aspect more in detail.

The technique used for obtaining the integer combinations in Q-OA actually results in an interpolation between the minimizer of the linear approximation in problem (OA-master) and the minimizer of the Lagrangean approximation, where  $\alpha$  in Eq. (2) is the interpolation parameter. Setting  $\alpha = 1$  will force the solution of problem (QOA-master) to the minimizer of problem (OA-master), and setting  $\alpha$  close to zero will allow the solution to be close to the minimizer of the Lagrangean approximation. The Q-OA method will, therefore, be less sensitive to the accuracy of the Lagrangean approximation, compared to the method in [13]. In the next section, we prove that finite convergence of Q-OA can still be guaranteed even if the Hessian of the Lagrangean is only estimated as long as it remains positive semidefinite.

To provide a geometric interpretation of the method and to show how it differs from OA and L-OA, we apply the method to the illustrative test problem (Ex 1). We use the same starting point  $(x^0, y^0)$  as before and we set the level parameter as  $\alpha = 0.5$ . To solve the problem with these parameters Q-OA requires three iterations. The first two iterations are shown in Fig. 4. In the third iteration, we are able to verify optimality after only solving the MILP subproblem, since we obtain  $LB^3 = f(\bar{x}, \bar{y})$ . From the figure, note that the reduction constraint given by Eq. (9), prevents the algorithm from taking a too short step in the first iteration, and the optimal solution is actually obtained in the first iteration. If the trial solution had only been chosen as the minimizer of the Lagrangean relaxation, it would have resulted in less progress per iteration. It should also be noted that not a single infeasible integer combination was encountered.



**Fig. 4** The figures illustrate the first two iterations needed to solve problem (Ex 1) with the Q-OA method. The dashed ellipses represent the contours of the approximated Lagrangean used as the objective in the MIQP subproblem and the red line shows the level constraint given by  $\mu \leq \hat{f}_k^*$ . The circular dots represent the best found solution so far, the squared dots represent the solutions obtained from the MIQP subproblem and diamond shaped dots represent the solutions obtained by one of the NLP subproblems

### 5 Convergence properties

Proving finite convergence of L-OA and Q-OA can be done similarly as for the original OA, and some of the results from [10,13] are directly applicable. Finite convergence can be proven as follows. We show that an infeasible integer combination obtained by L-OA or Q-OA will be cut off by the cuts generated according to Eq. (1) and therefore, this integer combination cannot be obtained in any future iteration. Next, we prove that a specific integer combination cannot be obtained twice with either method unless optimality is proven. The methods, therefore, obtain new integer combinations at each iteration, and since there are only a finite number of such combinations, the methods will converge in a finite number of iterations.

Convexity of the nonlinear functions is crucial since it ensures that no feasible integer solution is cut off by the cuts generated by L-OA or Q-OA and that problem (OA-master) gives a valid lower bound, as is stated in Lemma 1. The lemma and a proof is also found in [13].

**Lemma 1** *Solving problem (OA-master) yields a valid lower bound to the optimum of the MINLP problem.*

**Proof** From the first order convexity condition we know that for any convex differentiable function  $\phi(\mathbf{x}, \mathbf{y})$ ,

$$\phi(\mathbf{x}, \mathbf{y}) \geq \phi(\mathbf{x}^0, \mathbf{y}^0) + \nabla\phi(\mathbf{x}^0, \mathbf{y}^0)^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^0 \\ \mathbf{y} - \mathbf{y}^0 \end{bmatrix} \quad \forall(\mathbf{x}, \mathbf{y}), (\mathbf{x}^0, \mathbf{y}^0) \in D_\phi,$$

where  $D_\phi$  is the domain in which the function is convex. Therefore, the feasible region of the problem (MINLP) will be overestimated and the objective function will be underestimated at each iteration. □

In Theorem 2, we prove that L-OA and Q-OA always find new integer combinations as long as optimality is not guaranteed, which requires some intermediate results given in the following two lemmas.

**Lemma 2** *An infeasible integer combination  $\mathbf{y}^k$ , i.e., an integer combination such that problem (NLP-I) is infeasible, will be cut off by the cuts generated in L-OA and Q-OA.*

**Proof** It is proved in [13], that solving the feasibility problem and adding cuts for the active constraints will cut off  $\mathbf{y}^k$  from the search space. For more details see [13] Lemma 1 page 331.  $\square$

**Lemma 3** *If the lower bound is not equal to the upper bound, then there exists a solution to the MIQP subproblems in L-OA and Q-OA.*

**Proof** Due to convexity, the linear approximation problem (OA-master) is always feasible if the MINLP problem is feasible. The MIQP subproblem in both L-OA and Q-OA contains the same constraints as problem (OA-master) and the reduction constraint. Since  $\hat{f}_k^*$  is calculated according to Eq. (2), the solution to problem (OA-master) is a feasible solution to the MIQP subproblem. If problem (OA-master) is infeasible, the search will be terminated since it verifies that the MINLP is infeasible.  $\square$

**Theorem 2** *If the lower bound is not equal to the upper bound, then the MIQP subproblems in L-OA and Q-OA will give a new integer combination.*

**Proof** By Lemma 2, we know that any infeasible integer combination that has been found is cut off from the search space by the cuts added to the subproblems. Since the upper and lower bound are not equal, we know that the estimated optimum will be smaller than the upper bound, i.e.,  $\hat{f}_k^* < f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ . This is obviously true for all feasible solutions found so far, which we denote as  $\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i$ , and the following relation is obtained,

$$\hat{f}_k^* < f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq f(\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i) \quad \forall i. \quad (10)$$

At all the obtained feasible solutions  $\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i$ , the methods generate the following linearizations of the objective,

$$f(\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i) + \nabla f(\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i)^T \begin{bmatrix} \mathbf{x} - \hat{\mathbf{x}}^i \\ \mathbf{y} - \hat{\mathbf{y}}^i \end{bmatrix} \leq \mu. \quad (11)$$

In both the MIQP subproblem in L-OA and in Q-OA, we have the reduction constraint  $\mu \leq \hat{f}_k^*$ , and from Eq. (11) it follows that the next solution must satisfy,

$$\nabla f(\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i)^T \begin{bmatrix} \mathbf{x} - \hat{\mathbf{x}}^i \\ \mathbf{y} - \hat{\mathbf{y}}^i \end{bmatrix} < 0 \quad \forall i. \quad (12)$$

Now, assume that one of the feasible solutions  $\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \in \{\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i\}$  can be perturbed in the  $\mathbf{x}$ -variables by  $\Delta \mathbf{x}$  such that it satisfies all constraints of the MIQP subproblem and the property given by Eq. (12). Since  $\tilde{\mathbf{x}}$  was obtained by solving problem (NLP-I), it must satisfy the KKT-conditions,

$$\begin{aligned}
 \nabla_{\mathbf{x}} f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \sum_{j=1}^l \lambda_j \nabla_{\mathbf{x}} g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \mathbf{A}^T \boldsymbol{\gamma} &= \mathbf{0} \\
 g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) &\leq 0 \quad \forall j = 1, \dots, l \\
 \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{y}} &\leq \mathbf{b} \\
 \lambda, \boldsymbol{\gamma} &\geq \mathbf{0} \\
 \lambda_j g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) &= 0 \quad \forall j = 1, \dots, l \\
 (\mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{y}} - \mathbf{b}) \circ \boldsymbol{\gamma} &= \mathbf{0},
 \end{aligned}
 \tag{13}$$

where  $\nabla_{\mathbf{x}}$  is the gradient with respect to  $\mathbf{x}$ -variables, and  $\boldsymbol{\gamma}$  are the multipliers of the linear constraints. At the solution  $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ , the methods will generate the following supporting hyperplanes,

$$g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) + \nabla g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \begin{bmatrix} \mathbf{x} - \tilde{\mathbf{x}} \\ \mathbf{y} - \tilde{\mathbf{y}} \end{bmatrix} \leq 0 \quad \forall j \mid \lambda_j \neq 0.
 \tag{14}$$

Since these are all active constraints, the constant on the left-hand side must be zero, i.e.,  $g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = 0$ . The perturbation  $\Delta \mathbf{x}$  must satisfy Eq. (14), which can be written as,

$$\lambda_j \nabla_x g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} \leq 0 \quad \forall j = 1, \dots, l.
 \tag{15}$$

The same is also true for the linear constraints. For all active linear constraints  $\Delta \mathbf{x}$  cannot increase the value of the left hand side. This condition can be summed over all linear constraints by the multipliers  $\boldsymbol{\gamma}$  as

$$\boldsymbol{\gamma}^T \mathbf{A} \Delta \mathbf{x} \leq 0.
 \tag{16}$$

The perturbation also has to satisfy the reduction stated in Eq. (12), which yields,

$$\nabla_x f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} < 0.
 \tag{17}$$

Adding all inequalities from Eqs. (15), (16) and (17) results in the following strict inequality,

$$\nabla_x f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} + \sum_{j=1}^l \lambda_j \nabla_x g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} + \boldsymbol{\gamma}^T \mathbf{A} \Delta \mathbf{x} < 0.
 \tag{18}$$

However, taking the inner product of  $\Delta \mathbf{x}$  and both sides of the first KKT condition results in the following equality

$$\nabla_x f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} + \sum_{j=1}^l \lambda_j \nabla_x g_j(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^T \Delta \mathbf{x} + \boldsymbol{\gamma}^T \mathbf{A} \Delta \mathbf{x} = 0,
 \tag{19}$$



which leads to a contradiction. Therefore, there cannot exist a  $\Delta \mathbf{x}$  that satisfies all constraints of the MIQP subproblems without a change in the  $\mathbf{y}$ -variables. As stated in Lemma 3, there always exists a solution to the MIQP subproblems as long as the lower bound is not equal to the upper bound. Solving the MIQP subproblem will, therefore, result in a new integer combination different from all previously obtained solutions.  $\square$

Note that no assumptions were made in Theorem 2 regarding optimality of the MIQP subproblem. Therefore, the theorem is true for any solution that satisfies all constraints of the MIQP subproblem, optimal or not. Furthermore, Theorem 2 holds even if we make an arbitrary change to the objective function in the MIQP subproblems. An estimate of the Hessian in Q-OA will, therefore, be sufficient for Theorem 2 to hold. The next theorem summarizes the convergence properties.

**Theorem 3** *Both L-OA and Q-OA will terminate after a finite number of iterations, either by verifying optimality of the best-found solution or proving that the MINLP problem is infeasible.*

**Proof** From Lemma 1, it is clear that solving problem (OA-master) will either provide a valid lower bound or prove infeasibility. Furthermore, the proof of Lemma 1 also establishes that no feasible solution will be excluded from the search space. According to Theorem 2, both L-OA and Q-OA will find new integer combinations at each iteration as long as the gap between the upper and lower bound is not equal to zero. Since the linear constraints are assumed to give rise to a compact set, it is clear that there can only exist a finite number of different integer combinations, and thus, both methods must terminate after a finite number of iterations.  $\square$

Hence, we have proved that both proposed methods converge to a global optimal solution in a finite number of iterations. In the next section, we present a numerical comparison of the proposed methods and compare the results to the original OA method.

## 6 Computational results

In this section, we discuss our computational experiments and the obtained results. To compare the practical performance of the methods, we have implemented the original OA as well as L-OA and Q-OA. The main advantage of L-OA and Q-OA compared to the original OA, is the ability to handle highly nonlinear MINLP problems more efficiently. L-OA is more conservative when choosing the trial solutions, and tries to stay close to the best found feasible solution, which should reduce the number of infeasible integer combinations obtained. In Q-OA we are also able to incorporate second order information when choosing new integer combinations. Hence, the new integer combination is chosen with information regarding the curvature around the current solution. From the test problems, we observed a significant reduction in the number of iterations with both L-OA and Q-OA compared to the original OA.

To test and compare the methods we have implemented them and applied them to convex MINLP problems obtained from MINLPlib2 (rev. 373, as of 2017-11-07)<sup>1</sup> [15]. This set was chosen since it contains a large variety of different test problems originating from both practical applications as well as theoretical test problems. As mentioned earlier, both L-OA and Q-OA are intended for problems with high to medium degrees of non-linearity, and therefore, we used the following criteria for choosing the test problems

1. Classified as convex.
2. Having at least one discrete variable.
3. Having at least one continuous variable.
4. Satisfying the following inequality

$$\frac{n_{nonlin}}{n + m} > 0.5, \quad (20)$$

where  $n_{nonlin}$  is the number of variables present in some nonlinear term and  $m + n$  is the total number of discrete and continuous variables. There are in total 109 convex MINLP problems in MINLPlib2 (rev. 373, as of 2017-11-07) that satisfy the given criteria. These problems originate from several applications such as process synthesis, facility layout problems, batch design with storage, portfolio optimization and MINLP test problems. The test instances have between 7 and 4530 variables and 0–1822 constraints. More details of the test instances are provided in the supplemental material.

Next, we describe some details regarding the implementation of the methods and the computational results are presented in Sects. 6.2 and 6.3.

## 6.1 Implementation details

The implementation of the methods compared here was made in MATLAB using Gurobi 7.5.1 [19] as subsolver for the MILP/MIQP subproblems and IPOPT 3.12.7 [32] for the NLP subproblems. Furthermore, we use some functionality from OPTI Toolbox [7] to read the test problems.

Both L-OA and Q-OA require a feasible starting solution, and to obtain such a solution we start by performing a few original OA iterations. Once a feasible solution is obtained, we switch to either L-OA or Q-OA. The level parameter  $\alpha$  was set to 0.5 with both methods in the comparison.

According to Theorem 2, it is not necessary to find the optimal solution for the MIQP subproblems, and any feasible solution for these problems is sufficient for guaranteeing that both L-OA and Q-OA converge to the global optimum. This is an important property, since solvers such as Gurobi or CPLEX are often able to quickly find several feasible solutions, and quite often the majority of the solution time is spent proving optimality. We use a strategy of stopping the solver once a certain number of feasible solutions have been found, and specifically, we stop after 10 solutions have been found. This is simply done by setting the SolutionLimit parameter to 10. Using this

<sup>1</sup> <http://www.gamsworld.org/minlp/minlplib2/html/index.html>.

approach, we hope to ensure that we obtain a good solution to the MIQP problem, while significantly reducing the total solution time. For the MIQP subproblems, we always have a feasible solution available, the solution to the MILP subproblem (OA-master), and providing this as a starting solution to Gurobi also improved the performance. For the MILP subproblems, we used the default settings in Gurobi, and we also used the default settings for IPOPT.

The NLP subproblem (NLP-I) is always convex for these test problems. However, for some specific test problems we encountered some difficulties where the solver failed to find the optimal solution. Such difficulties could, for example, be caused by a specific integer combination not satisfying the constraint qualifications. These issues were not frequent and they only occurred for a few test problem in the entire MINLPLib2. To deal with such issues we chose a simple approach; if the NLP subproblem (NLP-I) is feasible but the NLP solver fails, we generate cutting planes for all violated constraints at the solution given by the MILP subproblem (OA-master) according to Eq. (1). These cuts will exclude the current solution to subproblem (OA-master) from the search space [35], and thus prevent cycling. Adding these cuts is equivalent to performing an iteration of the ECP method. From the convergence properties of the ECP method, we know that adding these cuts will eventually result in a new integer combination or verify optimality of an obtained solution.

Since the problems we consider are all convex, the Hessian of the Lagrangean is always positive semidefinite. However, due to numerical accuracy we did encounter a few cases where the Hessian was not strictly positive semidefinite, i.e., the smallest eigenvalue was not positive but in the range of  $-10^{-9}$ . To make sure that the MIQP subproblems are convex, we slightly modify the diagonal elements of the Hessian. For each row  $i$  of the Hessian which contains a nonzero element, we modify the diagonal by

$$\nabla_{\mathbf{x},\mathbf{y}}^2 \mathcal{L}(i, i) := \nabla_{\mathbf{x},\mathbf{y}}^2 \mathcal{L}(i, i) + |\lambda_{min}|, \quad (21)$$

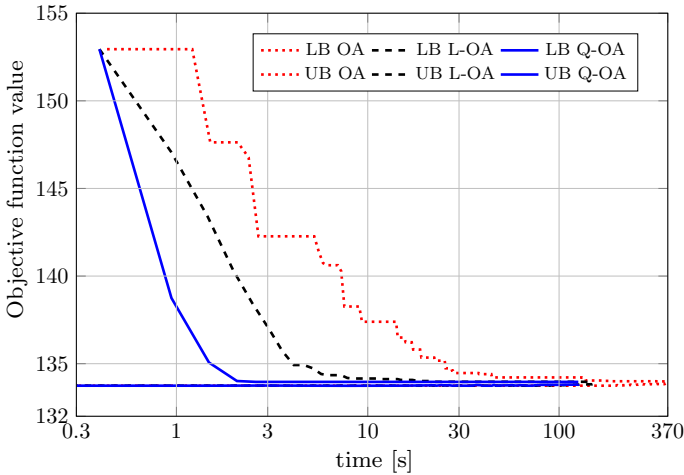
where  $\lambda_{min}$  is chosen as the smallest eigenvalue of the Hessian. This modification guarantees that all eigenvalues are positive [17], and thus, ensures convexity of the MIQP subproblem. The modification of the Hessian is only done in case one of the eigenvalues are negative.

As termination criteria, we used both an absolute optimality tolerance  $\epsilon$  and a relative optimality tolerance  $\epsilon_{rel}$ . The search is, thus terminated if either

$$f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - LB \leq \epsilon \quad \text{or} \quad \frac{f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) - LB}{|f(\bar{\mathbf{x}}, \bar{\mathbf{y}})| + 10^{-10}} \leq \epsilon_{rel}$$

are satisfied. Here  $LB$  denotes the current lower bound. These can be considered as the standard termination criteria for MINLP problems.

All tests were performed on an Intel Core i7 2.93GHz CPU desktop with 16GB of RAM running Windows 7. As termination criteria, we set the tolerances  $\epsilon = 10^{-5}$  and  $\epsilon_{rel} = 10^{-3}$ , and a time limit of 900 s.



**Fig. 5** Bound profiles for instance *cvxnonsep\_nsig40* against time. The figure shows the upper bound (UB) and lower bound (LB) obtained by the OA, L-OA, and Q-OA methods

## 6.2 Illustrative examples

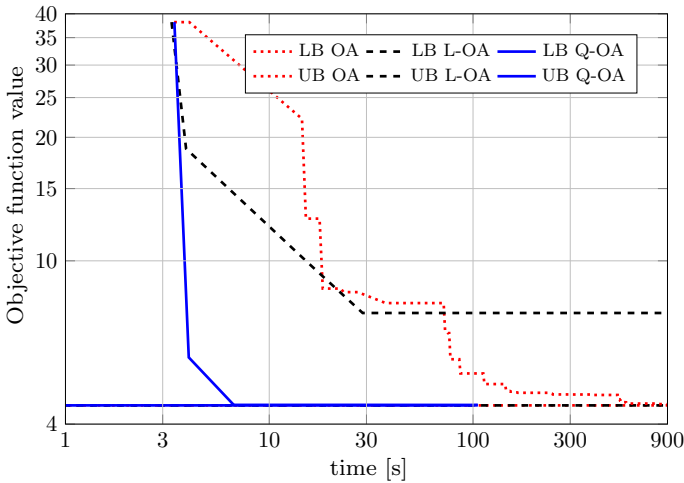
In this section, we present more detailed results of two particular instances of the selected test set. These instances were chosen such that they could exemplify the results shown in the following section. The selected instances are *cvxnonsep\_nsig40*<sup>2</sup> and *ibs2*.<sup>3</sup> The first instance was proposed by Kronqvist et al. [24], it contains 20 integer variables and 20 continuous variables, a linear objective and a signomial constraint. This seemingly simple problem is designed to be challenging for methods such as OA and ECP due to a highly nonlinear constraint. The second instance has 1500 binary variables, 1510 continuous variables, a linear objective, and 1821 constraints, of which 10 are nonlinear including square and logarithm operators. This problem represents a particular challenge for the OA method given its combinatorial complexity and the fact that most of its variables, discrete and continuous, are involved in a nonlinear fashion in the constraints.

To illustrate how the methods differ for these problems, we show the upper and lower bounds obtained by each method. Figures 5 and 6 show the progress of the bounds as a function of time for problems *cvxnonsep\_nsig40* and *ibs2*, respectively. From the figures, it can be observed that Q-OA is able to improve the upper bound more quickly than the other methods. This is usually the case and is explained by that fact that Q-OA utilizes more information when choosing the integer combinations than the other methods. Especially for the instance *ibs2*, there was a clear advantage of incorporating information from the second order derivatives, and Q-OA clearly performs better than L-OA.

From the results presented in the bounds profiles and in Table 1, we notice how including the level regularization can improve the performance of the OA method

<sup>2</sup> [http://www.gamsworld.org/minlp/minplib2/html/cvxnonsep\\_nsig40.html](http://www.gamsworld.org/minlp/minplib2/html/cvxnonsep_nsig40.html).

<sup>3</sup> <http://www.gamsworld.org/minlp/minplib2/html/ibs2.html>.



**Fig. 6** Bound profiles for instance `ibs2` against time. The figure shows the upper bound (UB) and lower bound (LB) obtained by the OA, L-OA, and Q-OA methods

while solving convex MINLPs. For the instance `cvxnonsep_nsig40` we notice a reduction in time of 59% and 66% using the L-OA and the Q-OA method, respectively. Although the new methods require the solution of an MIQP subproblem in each iteration, the extra time invested in finding the next integer combination is compensated with a reduction in both iterations and time.

For the instance `ibs2`, OA is unable to close the optimality gap under 0.1% within the time limit of 900 s even though it performs 431 iterations. When solving the problem with L-OA the upper bound initially diminishes faster in terms of both time and iterations compared to OA, but the MIQP subproblems become hard to solve resulting in only 41 iterations before hitting the time limit. When utilizing second order information with the Q-OA method, the problem is solved within an optimality gap of 0.1% in 104 s and just after 16 iterations, while only encountering 2 infeasible NLP subproblems. Note that, for both illustrative examples the methods are all able to obtain a tight lower bound already in the first iteration, which is not generally the case. Usually, the lower bounds can also vary significantly between the methods.

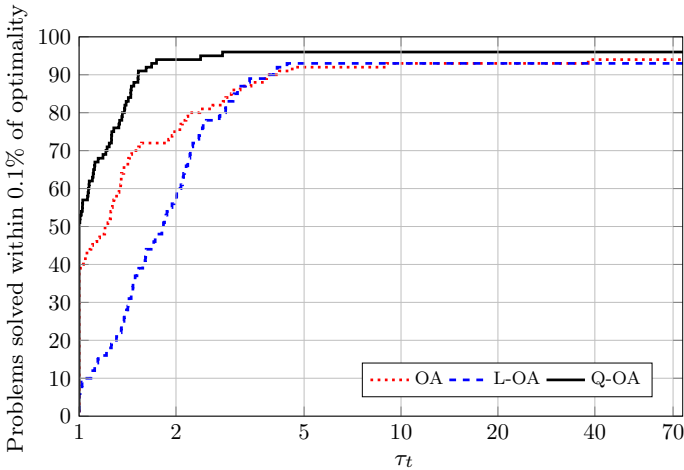
### 6.3 Numerical results

Having observed the improvement in performance of the proposed methods compared to OA in the illustrative examples, we considered the whole test set defined at the beginning of this section. In order to compare the performance of the methods, we have used performance profiles [9] both in terms of solution time and iterations in Figs. 7 and 8, respectively. The profiles show the number of problems solved against the respective performance ratio threshold  $\tau$ . A data point at each plot represents the number of instances that each method solved within a factor  $\tau$  of the best solver.

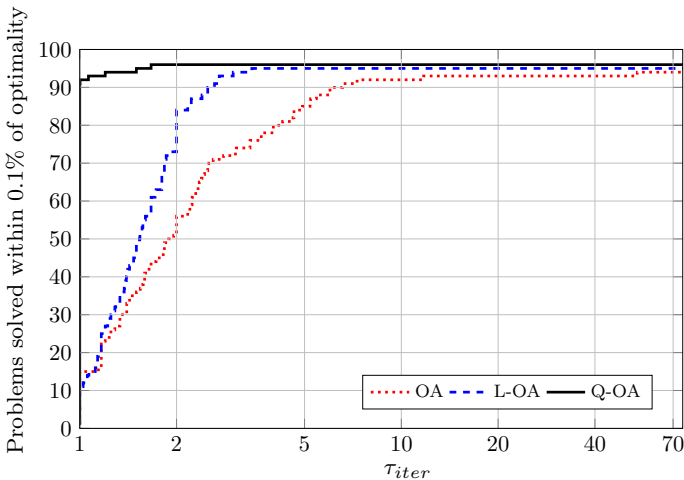
**Table 1** Detailed results of the illustrative examples while solving them with the OA, L-OA, and Q-OA methods

Instance	Solution method	Time (s)	Iterations	NLP time (s)	MILP time (s)	MIQP time (s)	Infeasible NLPs	Optimality gap
cvxnonsep_nsig40	OA	360.86	663	31.69	328.13	0	1	0.00098
	L-OA	147.87	201	9.84	55.35	82.37	0	0.000999
	Q-OA	121.55	144	6.65	38.68	75.99	0	0.000913
ibs2	OA	900*	431	152.86	747.14	0	6	0.005368
	L-OA	900*	41	40.16	17.98	841.86	6	0.1093
	Q-OA	103.89	16	48.60	6.469	48.68	2	0.000997

\*Time limit



**Fig. 7** Time performance profiles for test problems



**Fig. 8** Iterations performance profiles for test problems

Figure 7 shows how the Q-OA method is superior to both L-OA and OA for the selected test set in terms of solution time. The figure shows that Q-OA solves most instances to the desired optimality gap, and it solves the problems in the least amount of time. L-OA has initially the worst performance of the 3 methods for  $\tau \leq 3$ , but in the end, the performance is similar to that of OA without reaching the number of solved instances by Q-OA. It is also worth mentioning, that all the instances that remained unsolved with Q-OA are also unsolved with both OA and L-OA. Q-OA is thus able to solve all the problems solved with the other methods and some additional problems.

The performance profiles in terms of iterations in Fig. 8 show a clear advantage of Q-OA compared to the other 2 methods. Considering iterations, L-OA also performs

**Table 2** Number of instances solved and comparison in solution time and iterations of OA, L-OA, and Q-OA

Method	Instances solved	Less time than OA	Fewer iterations than OA	Less time than L-OA	Fewer iterations than L-OA	Less time than Q-OA	Fewer iterations than Q-OA
OA	94/109	–	–	61/94	23/94	38/94	3/94
L-OA	95/109	34/95	57/95	–	–	9/94	2/94
Q-OA	96/109	57/96	80/96	86/96	84/96	–	–

better than OA, and the profiles show a clear reduction in terms of iterations for both L-OA and Q-OA.

Given that the performance profiles show the results without distinguishing the individual instances, we include Table 2, which shows a direct comparison of the methods in terms of solution time and iterations. Note that Q-OA is able to solve 1 and 2 instances more than L-OA and OA, respectively.

None of the methods were able to find a solution within 0.1% of optimality gap for 13 instances in the test set. When comparing the proposed methods to OA, we see that L-OA is able to reduce the solution time in 36% of the instances and the iterations in 60%, while Q-OA reduced the time in 59% of the instances and the iterations in 83%. From the results, it was also noticed that the benefits of Q-OA are more apparent for the more challenging instances. By comparing the two proposed methods we see that Q-OA solves 90% of the instances in less time and 87.5% of the instances in fewer iterations than L-OA. Detailed solution information for all instances and methods can be found in the supplemental material.

An interesting result is that the proposed methods significantly decreased the number of infeasible NLP subproblems found while solving the selected problems. Using OA we obtained 877 infeasible NLP subproblems, while using L-OA and Q-OA we only obtained 259 and 257, respectively. This can be explained by the fact that the integer combinations are chosen closer in the search space to the best feasible solution, and information about the curvature is utilized with the proposed methods. Choosing an integer combination close to a feasible solution also results in trial solutions close to the feasible region, which resulted in fewer infeasible trial solutions.

The solutions reported here were obtained using the level parameter  $\alpha = 0.5$ . We performed several tests varying the value of  $\alpha$ , which resulted in significant changes for individual instances but rather insignificant when considering the whole test set. Overall,  $\alpha = 0.5$  gave us the best results for this set of test problems.

## 7 Conclusions and future work

We have presented two new methods for solving convex MINLP problems, based on a regularization technique and a second order approximation of the Lagrangean. We have proven that both methods converge to the global optimal solution in a finite number of iterations, and shown that the proofs hold even if the MIQP subproblem is only solved approximately. Both methods are mainly intended for problems with moderate to high degrees of nonlinearity, and for such problems, both methods performed better than



the original OA method. The new method called Q-OA required significantly fewer iterations than the original OA, and there was also a clear advantage in the solution time. The advantage is due to the fact that more information is utilized when choosing the integer combinations. The method L-OA uses a regularization technique which we showed is equivalent to using a trust region. The regularization prevents large jumps between iterations and tries to keep the trial solutions close to the feasible region, and for the test problems, it gave an advantage over the original OA. For the test problems, Q-OA performed better than the other methods, both with respect to the number of iteration and time. Furthermore, using Q-OA we were able to solve a larger percentage of the test problems within the time limit.

As future work we plan to implement the methods in a more efficient and flexible framework, e.g., within an MINLP solver like DICOPT [18] or as part of a Toolkit in an optimization modeling software such as Pyomo or JuliaOpt. It could also be worth to investigate a dynamic update of the level parameter  $\alpha$ . For example, it could be possible to adjust the parameter based on the current optimality gap. Another idea would be to investigate if the concepts used within L-OA and Q-OA could be effectively integrated within the framework of the NLP/LP based branch and bound algorithm presented in [28].

**Acknowledgements** Jan Kronqvist is grateful for the grants given by Walter Ahlström foundation, Svenska tekniska vetenskapsakademien i Finland, Tekniikan edistämissäätiö, TFIF and Waldemar von Frenckells stiftelse, which made the research visit at Carnegie Mellon University possible. David E. Bernal and Ignacio E. Grossmann would like to thank the Center Advanced Process Decision Making (CAPD) for its financial support. The authors would like to acknowledge the Dagstuhl Seminar 18081 on Designing and Implementing Algorithms for Mixed-Integer Nonlinear Optimization, where an early version of the results shown in this manuscript were presented and discussed.

## References

1. Bagirov, A., Karmitsa, N., Mäkelä, M.M.: Introduction to Nonsmooth Optimization: Theory, Practice and Software. Springer, Berlin (2014)
2. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. *Acta Numer.* **22**, 1–131 (2013). <https://doi.org/10.1017/S0962492913000032>
3. Biegler, L.T., Grossmann, I.E.: Retrospective on optimization. *Comput. Chem. Eng.* **28**(8), 1169–1192 (2004)
4. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. *Discr. Optim.* **5**(2), 186–204 (2008). <https://doi.org/10.1016/j.disopt.2006.10.011>
5. Bonami, P., Cornuéjols, G., Lodi, A., Margot, F.: A feasibility pump for mixed integer nonlinear programs. *Math. Program.* **119**(2), 331–352 (2009)
6. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
7. Currie, J., Wilson, D.I.: OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. In: Sahinidis, N., Pinto, J. (eds.) *Foundations of Computer-Aided Process Operations*. Savannah, Georgia (2012)
8. Dakin, R.J.: A tree-search algorithm for mixed integer programming problems. *Comput. J.* **8**(3), 250–255 (1965)
9. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program. Ser. B* **91**(2), 201–213 (2002). <https://doi.org/10.1007/s101070100263>
10. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36**(3), 307–339 (1986)

11. den Hertog, D., Kaliski, J., Roos, C., Terlaky, T.: A logarithmic barrier cutting plane method for convex programming. *Ann. Oper. Res.* **58**(2), 67–98 (1995)
12. de Oliveira, W.: Regularized optimization methods for convex MINLP problems. *TOP* **24**(3), 665–692 (2016)
13. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. *Math. Program.* **66**(1), 327–349 (1994)
14. Floudas, C.A.: Deterministic global optimization. Theory, methods and applications In: *Nonconvex Optimization and its Applications*, vol. 37. Springer, US (2000)
15. GAMSWorld: mixed-integer nonlinear programming library. <http://www.gamsworld.org/minlp/minplib2/html/> (2016). Accessed 24 Nov 2016
16. Geoffrion, A.M.: Generalized Benders decomposition. *J. Optim. Theory Appl.* **10**(4), 237–260 (1972)
17. Gershgorin, S.A.: Über die Abgrenzung der Eigenwerte einer Matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na* **6**, 749–754 (1931)
18. Grossmann, I.E., Viswanathan, J., Vecchietti, A., Raman, R., Kalvelagen, E.: *GAMS/DICOPT: A Discrete Continuous Optimization Package* (2002)
19. Gurobi Optimization, I.: Gurobi optimizer reference manual. <http://www.gurobi.com> (2016)
20. IBM Corp., IBM: V12.6: User's Manual for CPLEX. *Int. Bus. Mach. Corp.* **12**(1), 481 (2009)
21. Kelley Jr., J.E.: The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* **8**(4), 703–712 (1960)
22. Kiwiel, K.C.: Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Math. Program.* **69**(1–3), 89–109 (1995)
23. Kronqvist, J., Lundell, A., Westerlund, T.: The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *J. Glob. Optim.* **64**(2), 249–272 (2016)
24. Kronqvist, J., Lundell, A., Westerlund, T.: Reformulations for utilizing separability when solving convex MINLP problems. *J. Glob. Optim.* **71**(3), 571–592 (2018). <https://doi.org/10.1007/s10898-018-0616-3>
25. Lee, J., Leyffer, S. (eds.): *Mixed Integer Nonlinear Programming*, vol. 154. Springer, Berlin (2011)
26. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. *Math. Program.* **69**(1–3), 111–147 (1995)
27. Nesterov, Y.: *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer, Berlin (2004)
28. Quesada, I., Grossmann, I.E.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Comput. Chem. Eng.* **16**(10–11), 937–947 (1992)
29. Slater, M., et al.: *Lagrange multipliers revisited*. Technical report, Cowles Foundation for Research in Economics, Yale University (1959)
30. Trespalacios, F., Grossmann, I.E.: Review of mixed-integer nonlinear and generalized disjunctive programming methods. *Chem. Ing. Tech.* **86**(7), 991–1012 (2014). <https://doi.org/10.1002/cite.201400037>
31. Viswanathan, J., Grossmann, I.E.: A combined penalty function and outer-approximation method for MINLP optimization. *Comput. Chem. Eng.* **14**(7), 769–782 (1990)
32. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**(1), 25–57 (2006)
33. Wei, Z., Ali, M.M.: Outer approximation algorithm for one class of convex mixed-integer nonlinear programming problems with partial differentiability. *J. Optim. Theory Appl.* **167**(2), 644–652 (2015). <https://doi.org/10.1007/s10957-015-0715-y>
34. Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex MINLP problems. *Comput. Chem. Eng.* **19**, S131–S136 (1995)
35. Westerlund, T., Pörn, R.: Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optim. Eng.* **3**(3), 253–280 (2002)
36. Wolfe, P.: A duality theorem for non-linear programming. *Q. Appl. Math.* **19**(3), 239–244 (1961)
37. Zaourar, S., Malick, J.: Quadratic stabilization of Benders decomposition. <https://hal.archives-ouvertes.fr/hal-01181273> (2014). Working paper or preprint