A Tabu Search-based algorithm for the integrated process and control system design*

Oliver Exler†, Luis T. Antelo‡, José A. Egea‡, Julio R. Banga‡, Antonio A. Alonso‡

October 11, 2006

Abstract

The problem of integrated process and control system design is discussed in this paper. We formulate the optimization problem as a mixed-integer nonlinear programming problem subject to differential-algebraic constraints. This class of problems is frequently non-convex and local optimization techniques usually fail to locate the global solution. Thus, we propose a global optimization algorithm based on the metaheuristic Tabu Search to solve the challenging problem. The ideas of the methodology are explained and on the basis of two case studies the performance of the approach is evaluated. The first benchmark problem is a Wastewater Treatment Plant model [1] for nitrogen removal and the second case study is the well-known Tennessee Eastman Process [9]. Numerical experiments with our solver indicate that we can achieve an improved performance in both cases. Additionally, our solver outperforms several other solvers which we also used to solve the two case studies.

Keywords: Integrated Process and Control System Design, Mixed-Integer Non-Linear Programming (MINLP), Metaheuristic, Tabu Search, Tennessee Eastman Process.

1 Introduction

During the last decade, the importance of a simultaneous (integrated) process design approach, considering operability together with the economic issues, has been widely recognized (Morari and Perkins [17]; Pistikopoulos and Ross [20]; Bansal et al [5]; Sakizlis et al [21]). The aim is to obtain profitable and operable process and control structures in a

*This research was supported by the EU Marie-Curie Research Training Network under project number MRTN-CT-2004-512233

†University of Bayreuth, Department of Computer Science, 95440 Bayreuth - Germany

‡Process Engineering Group, IIM-CSIC, C/ Eduardo Cabello 6, 36208 Vigo - Spain

systematic way. Both the process design characteristics, control strategies, control structure and controller's tuning parameters have to be selected optimally in order to minimize the total cost of the system while satisfying a large number of feasibility constraints in the presence of time-varying disturbances.

The arising optimization problem is a challenging mixed-integer dynamic optimization (MIDO) problem. The multimodal (non-convex) nature of these problems has been highlighted by e.g. Schweiger and Floudas [22] and Bansal et al. [5]), among others. Consequently, the use of global optimization (GO) techniques is mandatory.

In the domain of deterministic GO methods, Esposito and Floudas [10] have presented approaches to solve dynamic optimization problems. This is indeed a very promising and powerful approach, but restrictions may apply for the type of path constraints which can be handled. Other groups (Singer et al [24]; Papamichail and Adjiman [19]) are also making good progress in deterministic global optimization of dynamic systems, yet several issues regarding requirements and computational performance are still present.

Regarding stochastic GO methods, a number of researches have shown that they can locate the vicinity of global solutions for nonlinear dynamic problems with relative efficiency (Banga et al [4]; Moles et al [16]; Sendin et al [23]), but the cost to pay is that global optimality can not be guaranteed. However, in many practical situations these methods can be satisfactory if they provide us with a *good enough* (often, the best available) solution in modest computation times. Furthermore, stochastic methods do not require transformation of the original problem, which can be treated as a black box. Thus, they can handle problems with complicated dynamics (e.g. discontinuities, non-smoothness, etc.). In our case we use the metaheuristic Tabu Search (TS) - often used successfully in the area of operations research (Glover and Laguna [12]) - for solving the integrated process and control system design problem.

This paper is structured as follows. The general statement of the problem is presented in section 2. Section 3 is dedicated to the optimization methodology that has been applied. In section 4 two challenging case studies are presented. The first benchmark problem is a Waste Water Treatment Plant model (see [1] and references therein) for nitrogen removal and the second case study is the well-known Tennessee Eastman Process (see [9]). We present the numerical results obtained by using our methodology, furthermore, we compare the performance of our approach with other solvers.

2 The integrated process and control system design problem

In order to represent the interaction of process design and control the formulation of the problem has to combine both components that express design alternatives and components that express the operability of the system. In general a superstructure is developed that contains all the possible alternatives that one wants to consider. Measuring the controllability of the process is achieved by introducing a system of differential and algebraic equations that simulates the behaviour of the process under dynamic operation.

Taking this into account the mathematical formulation is as follows:

$$\begin{array}{ll} \min_v & J(v, t_f) \\ s.t. & f(\dot{x}, x, p, v) = 0 \\ & x(t_0) = x_0 \\ & h(x, p, v) = 0 \\ & g(x, p, v) \geq 0 \\ & v^l \leq v \leq v^u \end{array} \tag{1}$$

where v is the vector of decision variables, J is the objective function (often representing the costs) to minimize, x is the state vector, f is the set of differential and algebraic equality constraints describing the system dynamics and h and g are possible equality and inequality path and/or point constraints which express additional requirements for the process performance. The lower and upper bounds for the decision variables are given with v^l and v^u.

As mentioned before a superstructure is developed that contains all alternatives for the process design. The common way to express alternatives in the design is introducing binary variables. In this way the active state of a design alternative can be easily expressed. A value equal to one stands for an active alternative whereas in the case of zero the alternative is inactive, i.e. not used. Thus, some of the elements of the decision variable vector v can be restricted to integer values. As a result the problem formulated in (1) is a mixed-integer dynamic optimization (MIDO) problem.

There are different approaches to solve this MIDO problem. They can be roughly classified in three groups, namely dynamic programming, control parameterization and complete discretization. The control parameterization technique parameterizes only the control variables, so the decision vector will contain that discretization information plus other selected time invariant parameters. The optimization is carried out in the space of these particular decision variables only. In our work we have followed this technique, i.e. we discretize the control variables and obtain as a result a finite dimensional mixed-integer non-linear programming (MINLP) problem with a dynamic system embedded, usually as a set of DAEs.

3 Description of the optimization method

Many algorithms for global optimization use a local solver to identify a local minimum by starting from an initial point and in order to reach the global minimum a special strategy for deciding where to start the local solver is applied. For convex problems the local solver is able to find the global minimum. If the problem is not convex the global optimality of the solution of the local solver cannot be guaranteed. In this case one has to guide the local solver to the global optimum. This second component of the algorithm has to locate the attraction basin of the global minimum so that a run of the local solver started in this basin will find the global minimum. The developed approach uses this strategy. As the global component a Tabu Search (TS) algorithm is applied. The local solver is a

special adaptation of a sequential quadratic programming method for the mixed-integer case. The following subsection is dedicated to the description of the interaction of the two different components and the basic ideas behind the proposed methodology, whereas the local solver, called MISQP [11], is described in a separate subsection.

3.1 The hybrid strategy - MITS

TS is a metaheuristic originally developed by Glover (see Glover et al. [12]). For the optimization of combinatorial problems TS has been tested very successful. In the last years it also has been applied to the optimization of continuous problems. An adaptation of TS to mixed-integer nonlinear optimization problems, called Mixed-Integer Tabu Search (MITS) is presented. The presented algorithm is an enhancement of an approach proposed by Battiti and Tecchiolli [6]. First, let us summarize the basic idea of a tabu search algorithm. The algorithm starts from an initial solution v_0. For this current solution v_k a set of neighbors is generated and the best among these neighbors is chosen to be the next iteration point, even if the function value is worse than the one of the current iterate v_k. Allowing an increase in the objective function is necessary to escape from a local minimum. To avoid cycling and to guide the search into unexplored areas, some former visited points are set to be tabu and so prohibited for some time. This procedure is repeated by starting from the new current point, until some stopping condition is fulfilled.
The efficiency of the above mentioned procedure depends on the choice of some key parameters, i.e., the length of the period a point is set to be tabu. Especially in the case if continuous variables have to be handled there are several of these parameters which have to be set right a priori. Since we do not know anything about the surface of the functions, this is not desirable and we want to use a procedure that is independent from such choices. That is the main reason why we decided to use the basic concepts from Battiti and Tecchiolli [6]. They proposed a TS algorithm that is robust for any kind of functions and self-adjusting, so that no parameters have to be set a priori. The components will be described now in more detail.

3.1.1 Neighborhood

In the basic tabu search for discrete optimization the neighborhood of an iteration point is built by all the direct neighbors of this point. Since we have to handle large mixed integer problems, it is impossible to consider all neighbours and it is even a hard task to define exactly what a direct neighbor is. Thus, we have to restrict the neighborhood to some generated points. The idea is to subdivide the search space into regions that represents basins of attraction and to perform the search over these regions. As mentioned before we created a neighbor generating routine that is self-adjusting and independent from parameters that have to be set a priori.
The initial search region is specified by bounds on each independent variable v_i: $v_i^l \leq v_i \leq v_i^u$, for $i = 1,\ldots,n$. Initially, this search region is subdivided into 2^n equal-sized regions, obtained by dividing the search space in half the range on each variable. This first guess

for the different regions is very crude but will be updated during the search. Each region is subdivided into 2^n smaller equal-sized regions as soon as two different local minima are found in it. Because of the splitting a region can contain at most one local minimum. In this way the search will react on the local properties of the objective function. The regions will be stored internally as a tree and after some iteration of MITS the tree will be of varying depth in the different regions, with regions of smaller sizes representing regions that require an intensification of the search. The leaves of the tree partition the initial search region, where the intersection of two leaves is empty and the union of all leaves represents the initial search space.
In each iteration MITS generates a set of points, called the neighborhood $N(v_k)$ of the current iterate v_k. The iterate v_k lies in a unique region that can be identified by a unique binary string $B = [b_{11}, \ldots, b_{1N}, \ldots, b_{n1}, \ldots, b_{nN}]$. The value N represents the depth of the region in the tree. Initially all region are of depth one and after splitting a box the new smaller region have a depth increased by one compared to their father region. The length of the region edge along the i-th coordinate is therefore equal to $(v_i^u - v_i^l)/2^N$. The position of the origin B_i^N of the region containing the current iterate along the i-th coordinate is

$$B_i^N = v_i^l + (v_i^u - v_i^l) \sum_{j=1}^{N} b_{ij}/2^j. \tag{2}$$

The neighborhood $N(v_k)$ is a set of randomly generated points in the neighbored regions. Depending on the depth of the current region up to $3 \times n$ neighbors are created randomly in the corresponding regions. The number can vary according to the size of the current region. The origins of the neighbored regions can be obtained as follows. In each coordinate up to 2 neighbored regions are considered. Only the regions that do not violated the box constraints for the variables are evaluated. The origins of these neighbor regions are obtained by changing only in the i-th coordinate according to the rule

$$neighB_i^N = B_i^N \pm (v_i^u - v_i^l)/2^N. \tag{3}$$

The length of the neighbor region edge once again is $(v_i^u - v_i^l)/2^N$. In this way up to $2 \times n$ neighbors are created. Additionally, we start from the region of depth 1 that contains the current iterate that is represented by $B = [b_1, \ldots, b_n]$. From this box we generate n regions where the origins are calculated as follows

$$neighB_i^1 = B_i^1 \pm (v_i^u - v_i^l)/2. \tag{4}$$

Among these neighbors the best one is chosen and the next iteration starts from this point and the corresponding region. For the unconstrained case that is

$$v_{k+1} \in N(v_k) : J(v_{k+1}) = \min_{v \in N(v_k)} J(v) \tag{5}$$

where v is not tabu because it lies in a prohibited region.

3.1.2 Diversification vs. Intensification

An important question arising in global optimization is when to intensify the search in a specified region and when to explore new regions. The risk is to waste a lot of time and evaluations of the objective function in a region that is not containing the optimum or to miss the global optimum if the search was not intensified. In our approach the TS component has the purpose of diversifying the search and of exploring the whole search space whereas the local solver intensifies the search in a promising area found by the TS component and finds the local minimum with high precision. This section specifies how this goal is accomplished.
The diversification is done by generating the neighbors in the whole search space. In this manner every minimum can be reached. To enforce diversification we also prohibit recently visited regions. If the objective function or the constraint violation for the infeasible case of the current iterate v_k is less than all values of the generated neighbors, the local solver MISQP might be started. To start MISQP

$$J(v_k) \leq J(v) \; for \; all \; v \in N(v_k) \tag{6}$$

is a necessary condition, but it is not sufficient unless for the first time a point in the region containing v_k is locally optimal. In this case MISQP will always be executed. Otherwise, a new run of the local solver has to be justified. Battiti and Tecchiolli [6] apply some Bayesian stopping rules for multi-start global optimzsation to determine if a new run should be executed. We use the same strategy for MITS and so we summarize the idea in the following paragraph.
Let $r > 1$ be the number of times MISQP has been started in the current region. An additional MISQP run should be started if and only if there is a high probability of finding a new local minimum in the region. Starting MISQP in some parts of the region might lead the solver in a neighbored region. The region can be partitioned into W components, the attraction basins of the local minima contained in the region and the possible basins that lead MISQP outside. The probabilities of the basins can be summed up to one ($\sum_{w=1}^{W} P_w = 1$). If $r > W + 1$ restarts have been executed and W different components have been identified, the total relative volume of the observed region (i.e., the posterior expected value of the relative volume Ω) can be estimated by

$$E(\Omega|r, W) = \frac{(r - W - 1)(r + W)}{r(r - 1)}; \; r > W + 1. \tag{7}$$

MISQP is always triggered if $r \leq W + 1$. Otherwise, MISQP is executed again with probability equal to $1 - E(\Omega|r, W)$, that is if

$$rand > E(\Omega|r, W) \tag{8}$$

where *rand* is randomly generated number in the range $[0, 1]$. In this way, the number of MISQP runs can be kept down if the above estimate predicts a small probability of finding a new local minimum, but a new run is never completely prohibited for the reasons of

robustness.
The initial point for MISQP is the current iterate v_k and the initial search region is the current region enlarged by twice the range of the region. If v^* is a local minimum that was found by MISQP, it is saved in a memory structure associated to the region. If MISQP converges to a point outside the original region the solution will be the next starting point for the next iteration; otherwise, MITS continues from the best neighbor found in this iteration.
As mentioned before, as soon as two different local minima v^* and v^{**} are identified in a region, the current region is subdivided into 2^n equal-sized smaller regions. If v^* and v^{**} belong to two different regions of the new partition, the splitting is terminated; otherwise the splitting is applied to the region containing v^* and v^{**} until their separation is obtained. The local minima are associated with their new regions.

3.1.3 Tabu Tenor Update

To guide the search into unexplored regions some of the previous visited regions are set to be tabu, i.e. as soon as a region contains the current iterate v_k it is prohibited for the next iteration. Changing the tabu tenor T - the time a region is tabu and cannot be revisited - the algorithm can manage the interaction between intensification and diversification. The following basic rules are used:

- As soon as a region are visited again the tabu tenor is increased and the algorithm is forced to starts some diversification. In this way cycling should be avoided.
- When repetitions are absent for a sufficiently long period, the tabu tenor is reduced and the diversification will disappear.
- When the described tabu tenor changes are not sufficient and cycles still occur an escape mechanism is needed. This mechanism will force the search in new regions in a very drastic way.

Since the number of generated neighbors depends on the size of the current region our prohibition rule has to be flexible. The tabu tenor $T_f \in [0, 1]$ in our case regulates the fraction of prohibited regions in the possible set of neighbored regions. In detail, all neighbored regions that have been visited in the last

$$T = \begin{cases} \max\left(1, \min\left(\lfloor T_f n \rfloor, n-2\right)\right) & if\ N = 1 \\ \max\left(1, \min\left(\lfloor T_f 2n \rfloor, 2n-2\right)\right) & otherwise \end{cases}$$

iterations are tabu and cannot be visited. If $N > 1$ up to $3 \times n$ neighbors can be generated, but since sometimes only $2 \times n$ neighbors can be generated - additional regions would lie outside the box constraints - we use the formula $\lfloor T_f 2n \rfloor$. It might happen that this value is too small but this is feasible because the escape mechanism will be activated and avoid these cycles. In the normal situation, the list is proportional to $\lfloor T_f n \rfloor$ (or $\lfloor T_f 2n \rfloor$), the

max operator ensures that the list is at least one, the *min* operator ensures that it is at most $n-2$ (or $2n-2$). In this way, the last visited region is always prohibited, and at least two regions are available from a given point, so that the chosen move is influenced by the values of the objective function f - for the unconstrained case - in the neighborhood.

3.1.4 Aspiration Criterion

While the search is proceeding, several regions of the search space are classified as tabu. In some cases, the best neighbor solution may lie in a tabu area nut its objective function value is better than the current best value. In this situation the tabu property can be invalidated and the point will be chosen. This feature is necessary to enforce faster convergence to the global minimum.

3.1.5 Stopping conditions

Defining a stopping criterion for GO algorithms is very difficult. If the algorithm stops too early, the global optimum can be missed. Otherwise, if it starts too late, computational effort will be wasted. Our procedure will stop if one of the following criteria is fulfilled:

- The search procedure will stop after a predefined maximum number of iterations.
- The procedure will stop after a given number of iterations without any further improvement on the value of the objective function.
- The program will terminate after the predefined maximum time has elapsed.
- The program will also terminate after the predefined maximum number of function evaluations has elapsed.

Since we have to handle computational expensive models, time will be the bottleneck. Hence in most cases we will stop our algorithm after a maximum time has elapsed.

3.2 The local solver MISQP

The purpose of this section is to give a short overview of the mathematical theory of the local solver that is activated by MITS. The solver called MISQP (Mixed-Integer Sequential Quadratic Programming) is a SQP Trust-Region method developed by Exler and Schittkowski [11]. We consider the general optimization problem to minimize an objective function f under nonlinear equality and inequality constraints,

$$
\begin{aligned}
& \min f(x,y) \\
& g_j(x,y) = 0\ , \quad j = 1,\ldots,m_e\ , \\
x \in \mathbb{R}^{n_c}, y \in \mathbb{Z}^{n_i}: \ & g_j(x,y) \geq 0\ , \quad j = m_e+1,\ldots,m\ , \\
& x_l \leq x \leq x_u \quad , \\
& y_l \leq y \leq y_u \quad ,
\end{aligned}
\tag{9}
$$

where x denotes the vector of the continuous and y the vector of the integer variables. It is assumed that the problem functions $f(x,y)$ and $g_j(x,y)$, $j = 1, \ldots, m$, are continuously differentiable subject to $x \in \mathbb{R}^{n_c}$. We define $n = n_c + n_i$ and $g = (g_1, \ldots, g_m)^T$.

MISQP is an algorithm where we try to approximate the Lagrangian

$$L(x,y,u) := f(x,y) - \sum_{j=1}^{m} u_j g_j(x,y), \tag{10}$$

where $u = (u_1, \ldots, u_{m+n+n})^T \in \mathbb{R}^{m+n+n}$ is the multiplier vector, subject to the continuous and the integer variables by a quadratic function. The Hessian of the Lagrangian function - we call it B - is approximated by a quasi-Newton update formula subject to the continuous and integer variables. In our implementation the well-known BFGS-Formula is used. Since we do not assume that the MINLP is relaxable, i.e., that $f(x,y)$ and $g_j(x,y)$, $j = 1, \ldots, m$, can be evaluated at any fractional parts of the integer variables, we approximate the first derivatives at $f(x,y)$ by the difference formula

$$d_y f(x,y) = \frac{f(x, y_1, \ldots, y_j + 1, \ldots, y_{n_i}) - f(x, y_1, \ldots, y_j - 1, \ldots, y_{n_i})}{2}, \tag{11}$$

for $j = 1, \ldots, n_i$, at neighbored grid points. If either $y_j + 1$ or $y_j - 1$ violates a bound, we apply a non-symmetric difference formula. Similarly, $d_y g_j(x,y)$ denote a difference formula for first derivatives at $g_j(x,y)$ computed at neighbored grid points. For the continuous variables the gradients are numerically approximated by a forward difference formula. Instead of a line search as usually applied in the continuous case, we use trust regions to stabilize the algorithm and to enforce convergence. We proceed from the continuous trust region method of Yuan [27] with second order corrections.

The algorithm uses the exact penalty function

$$P_\sigma(x,y) := f(x,y) + \sigma \left\| g(x,y)^- \right\|_\infty \quad , \tag{12}$$

with a penalty parameter $\sigma > 0$ and

$$g_j(x,y)^- := \begin{cases} g_j(x,y) \quad , & \text{if } j \le m_e \quad , \\ \min(0, g_j(x,y)) \quad , & \text{otherwise} \quad . \end{cases} \tag{13}$$

In every iteration the mixed integer quadratic problem (MIQP)

$$d \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} : \begin{array}{l} \min \frac{1}{2} d^T B_k d + \nabla f(x_k, y_k)^T d + \sigma_k \left\| \left(\nabla g(x_k, y_k)^T d + g(x_k, y_k) \right)^- \right\|_\infty \\ \|d\|_\infty \le \Delta_k \quad . \end{array} \tag{14}$$

has to be solved. Since the generated subproblems are always convex, we apply a branch-and-bound algorithm by Spickenreuther [25] to solve this problem. The solution d_k of the quadratic subproblem (14) is used to compute the next iteration point

$$(x_{k+1}, y_{k+1}) = \begin{cases} (x_k, y_k) + d_k \quad , & \text{if } \; P_{\sigma_k}((x_k, y_k) + d_k) \le P_{\sigma_k}(x_k, y_k) \\ (x_k, y_k) \quad , & \text{otherwise} \quad . \end{cases} \tag{15}$$

A key role in the trust region algorithm plays the prediction of a new trust region radius for the next iteration. The trust region radius is adjusted according to the quotient of the actual and the predicted improvement of the merit function:

$$r_k := \frac{P_{\sigma_k}(x_k, y_k) - P_{\sigma_k}((x_k, y_k) + d_k)}{\phi_k(0) - \phi_k(d_k)} \quad , \tag{16}$$

where

$$\phi_k(d) := \frac{1}{2} d^T B_k d + \nabla f(x_k, y_k)^T d + \sigma_k \left\| \left(\nabla g(x_k, y_k)^T d + g(x_k, y_k) \right)^- \right\|_\infty . \tag{17}$$

is used to estimate the linearly predicted improvement, i.e., the objective function of subproblem (14).

The trust region radius Δ_k has to be updated in order to enforce convergence. If r_k is close to one or even greater than one, then Δ_k is enlarged and if r_k is very small, Δ_k is decreased. If r_k remains in the intermediate range, Δ_k is not changed at all. More formally, we use the same constants proposed by Yuan [27], and set

$$\Delta_{k+1} = \begin{cases} \max\left[2\Delta_k,\ 4\|d_k\|_\infty\right] , & \text{if} \quad r_k > 0.9 \ , \\ \Delta_k \ , & \text{if} \quad 0.1 \le r_k \le 0.9 \ , \\ \min[\Delta_k/4,\ \|d_k\|_\infty/2] \ , & \text{if} \quad 0 < r_k < 0.1 \ . \end{cases} \tag{18}$$

If, on the other hand, $r_k < 0$, then Δ_k is decreased and we solve subproblem (14) again.

To increase the convergence rate a second quadratic problem is used

$$d \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} : \begin{array}{l} \min \bar{\phi}_k(d) \\ \|d + d_k\|_\infty \le \Delta_k \ . \end{array} \tag{19}$$

where

$$\begin{aligned} \bar{\phi}_k(d) := {} & \tfrac{1}{2}(d + d_k)^T B_k (d + d_k) + \nabla f(x_k, y_k)^T (d + d_k) \\ & + \sigma_k \left\| \left(\nabla g(x_k, y_k)^T d + g((x_k, y_k) + d_k) \right)^- \right\|_\infty . \end{aligned} \tag{20}$$

Let the solution be $\hat{d}_k$. In order to state the algorithm we define the following two equations:

$$\bar{r}_k := r_k + \frac{\bar{\phi}_k(0) - \bar{\phi}_k(\hat{d}_k)}{\phi_k(0) - \phi_k(d_k)} \quad , \tag{21}$$

and

$$\hat{r}_k := \frac{P_k(x_k, y_k) - P_k((x_k, y_k) + d_k + \hat{d}_k)}{\phi_k(0) - \phi_k(d_k)} \quad . \tag{22}$$

In Algorithm 3.1 the formulation of the loacl solver MISQP is presented.

Algorithm 3.1 *(MISQP)*

0: Let $(x_1, y_1) \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i}$, $\Delta_1 > 0$, $B_1 \in \mathbb{R}^{n \times n}$ *positive definite,* $\delta_1 > 0$, $\sigma_1 > 0$, $\epsilon > 0$, *and let* $k := 1$.

1: Solve subproblem (14) to get d_k and the multiplier u_k. If $\phi_k(0) - \phi_k(d_k) < \epsilon$ and $g(x_k, y_k)^- < \epsilon$, then stop. Update penalty parameter σ_k.

2: Compute r_k by (16). If $r_k > 0.75$, goto Step 5. Solve subproblem (19) to get $\hat{d}_k$ and the corresponding multiplier $\hat{u}_k$, and compute $\bar{r}_k$ by (21). Let $u_k := \hat{u}_k$. If $r_k < 0.25$, goto Step 3. If $0.9 < \bar{r}_k < 1.1$, let $\Delta_{k+1} := 2\Delta_k$, else $\Delta_{k+1} := \Delta_k$ and goto Step 6.

3: If $\bar{r}_k < 0.75$, goto Step 4. Otherwise, compute $f((x_k, y_k) + d_k + \hat{d}_k)$ and $g((x_k, y_k) + d_k + \hat{d}_k)$. If $P_k((x_k, y_k) + d_k + \hat{d}_k) \geq P_k((x_k, y_k) + d_k)$, goto Step 4. Calculate $\hat{r}_k$ by (22) and let $d_k := d_k + \hat{d}_k$, $r_k := \hat{r}_k$. If $r_k \geq 0.75$, goto Step 5. If $r_k \geq 0.25$, goto Step 6.

4: Let $\Delta_{k+1} := \|d_k\|_\infty / 2$ and goto Step 6.

5: If $\|d_k\|_\infty < \Delta_k$, then $\Delta_{k+1} := \Delta_k$ and goto Step 6. If $r_k > 0.9$, then $\Delta_{k+1} := 4\Delta_k$, else $\Delta_{k+1} := 2\Delta_k$.

6: If $r_k > 0$, goto Step 7. Otherwise, let $(x_{k+1}, y_{k+1}) := (x_k, y_k)$, $B_{k+1} := B_k$, increment k, and goto Step 1.

7: Define a new iterate $(x_{k+1}, y_{k+1}) := (x_k, y_k) + d_k$, compute $f(x_{k+1}, y_{k+1})$ and $g(x_{k+1}, y_{k+1})$, update B_{k+1} by the BFGS formula applied to d_k and $\nabla L(x_{k+1}, y_{k+1}, u_k) - \nabla L(x_k, y_k, u_k)$. Increment k and goto Step 1.

4 Case Studies

We tested the tabu search implementation MITS on two well-known benchmark problems. In the first part we show the behavior of the algorithm in the case of a wastewater treatment plant (WWTP) and the second one is the Tennessee-Eastman process (TEP). The algorithm is implemented in Matlab. An interface is used to call the local solver MISQP that is implemented in FORTRAN.

4.1 A wastewater treatment plant

The first benchmark problem that we consider is a wastewater treatment plant (WWTP) for nitrogen removal. The WWTP was developed by the COST 624 work group [8] and additional information can be found in Copp [7] and Alex et al. [1]. The layout of this benchmark plant - as shown in Figure 1 - combines nitrification with predenitrification by a five-compartment reactor with an anoxic zone. A secondary settler separates the microbial culture from the liquid being treated. A basic control strategy consisting of 2 PI controllers is proposed to test the benchmark. Its aim is to control the dissolved oxygen level in the final compartment of the reactor (AS Unit 5) by manipulation of the oxygen transfer, and to control the nitrate level in the last anoxic compartment (AS Unit 2) by manipulating the internal recycle flow rate.

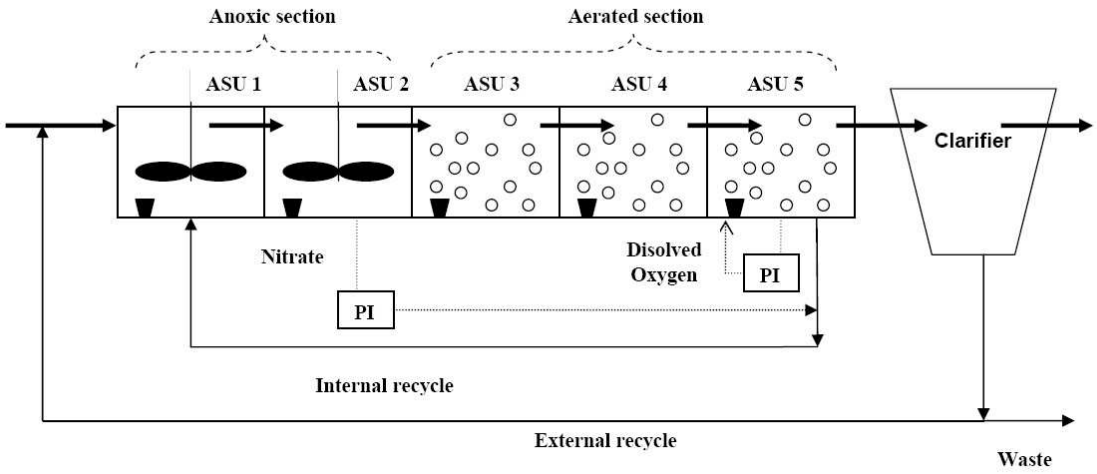

Figure 1: The Wastewater Treatment Plant

A Simulink implementation of the benchmark model by Jeppsson [14] was used for the simulations. Each function evaluation consists of an initialization period of 100 days to achieve steady state, followed by a period of 14 days of dry weather and a third period of 14 days of rainy weather. Calculations of the controller performance criterion are based on data from the last 7 rain days. Each simulation of this benchmark model takes a significant time on a standard PC (about 60 seconds in a PC with Intel Pentium IV 3,2 GHz).

The control performance is tested by using the ISE (Integral Square Error) as a controller performance criterion. Both the nitrate level and oxygen level controllers (further referred as N- and O-controller respectively) are optimized with respect to their controller parameters, that is, the gain K, integral time constant τ_i and anti-windup time constant τ_t. Besides the controller performance our objective function also reflects the following data:

- EQ: Effluent quality in $kg\ pollution\ units/d$ includes number of violations and of time in violation. Limits for each component are available.
- PE: average pumping energy, in kWh/d
- AE: average aeration energy, in kWh/d
- P_{Sludge}: sludge production

The problem is formulated as follows:

$$\begin{array}{ll} \min_v & J(v,t_f) = 2 \cdot EQ + PE + AE + 3 \cdot P_{Sludge} + 1000 \cdot (\omega_1 \cdot ISE_{(N)} + \omega_2 \cdot ISE_{(0)}) \\ s.t. & \dot{x} = f(\dot{x}, x, p, v) = 0 \\ & x(t_0) = x_0 \\ & v^l \leq v \leq v^u \end{array} \tag{23}$$

where $f \in \mathbb{R}^{150}$ denotes the system dynamics and $v \in \mathbb{R}^{12} \times \mathbb{Z}$ is the vector of decision variables. The system dynamics are described by algebraic mass balance equations, ordinary differential equations for the biological processes in the bioreactors as defined by the ASM1-model [13], and the double-exponential settling velocity function presented in [26] as a fair presentation of the settling process, with $x \in \mathbb{R}^{13}$ the state vector, $p \in \mathbb{R}^p$ the system parameters and d the influent disturbance. Due to the complexity of the system dynamics, the problem cannot be solved analytically, and the optimal values for the decision variables have to be retrieved by optimization techniques or open loop controller tuning.

The weights in the objective function are chosen to express the multi-objective character of the problem in a single objective function. By setting the weights to these values we achieve for each component the same magnitude in the computed values at the initial point. It should be highlighted that the values can be varied in order to change the weighting of different components, e.g., increasing the weight for the pump energy the optimization will lead to a smaller value in the pump energy.

The weights ω_1 and ω_2 are chosen such that in the initial point the equation

$$control = \omega_1 \cdot ISE_{(N)} + \omega_2 \cdot ISE_{(0)} = 1 \tag{24}$$

is satisfied, where

$$ISE_{(.)} = \int_{t_0}^{t_f} \epsilon(\tau)^2_{(.)} \, d\tau \ . \tag{25}$$

The exact values of ω_i, $i = 1, 2$, are negligible since the only purpose of the normalization is to simplify the comparison of controllability for different configurations.

Table 1 lists all decision variables with a short description and the upper and lower bounds. Boundaries on the decision variables (v^l and v^u) are chosen such that the process dynamics would not show (exceptional) unstable behavior.

As mentioned before, the weights in the objective function are chosen with respect to the performance obtained with default values for the decision variables (i.e. the tuned PI-parameters) provided by the COST project [8]. The optimization was started with the default values for the variables. Since the evaluation of the objective function is costly in the sence of time consuming we stopped MITS after exceeding a time limit of 2 days. Table 2 shows the default values and the values obtained by MITS.

The new configuration obtained by our solver leads to several changes, which are shown in Table 3. The performance has improved in all points except the average pumping energy and the sludge production. This behavior can be explained with the multi-objective nature of the objective function and the conflicting nature of the different criteria. As already

Table 1: Decision variables for the COST benchmark

Variable	Description	$[v^l\ v^u]$
v(1)	N-controller - gain	[100 1000]
v(2)	N-controller - integral time constant	[0.0007 0.7]
v(3)	N-controller - anti-windup time constant	[0.0001 0.7]
v(4)	O-controller - gain	[100 50000]
v(5)	O-controller - integral time constant	[0.01 1]
v(6)	O-controller - anti-windup time constant	[0.0001 0.07]
v(7)	Aeration factor in reactor 1	[0 360] d^{-1}
v(8)	Aeration factor in reactor 2	[0 360] d^{-1}
v(9)	Aeration factor in reactor 3	[0 360] d^{-1}
v(10)	Aeration factor in reactor 4	[0 360] d^{-1}
v(11)	Sludge purge	[0 1844.6] m^3/d
v(12)	Sludge recycle from settler	[0 36892] m^3/d
v(13)	Feed layer in settler $v(13) \in \mathbb{Z}$	[1 10]

Table 2: Result for the WWTP

Variable	Default value	Result MITS
v(1)	500	486.3
v(2)	0.001	0.000708
v(3)	0.0002	0.0001
v(4)	15000	16369.3
v(5)	0.05	0.0221
v(6)	0.03	0.0275
v(7)	0	0
v(8)	0	0
v(9)	10 h^{-1} (240 d^{-1})	224.8 d^{-1}
v(10)	10 h^{-1} (240 d^{-1})	224.8 d^{-1}
v(11)	385 m^3/d	333.1 m^3/d
v(12)	18446 m^3/d	16921 m^3/d
v(13)	5	7

Table 3: Changes in the plant performance

Criterion	Default	MITS	Unit
J - Objective function value	35226	33537	
EQ - Effluent Quality Index	9032	8356	$kg\ poll - units/day$
AE - Average Aeration Energy	7173	6787.6	kWh/day
PE - Average Pumping Energy	1919	2045.7	kWh/day
P_{Sludge} - Sludge Production	2357	2393.8	m^3/day
$ISE_{(O)}$	$1.2232 \cdot 10^{-5}$	$2.1779 \cdot 10^{-4}$	$(mg\ -COD/l)^2 \cdot day$
$ISE_{(N)}$	0.83345	0.46733	$(mgN/l)^2 \cdot day$
$control$	1.0	0.81	

stated before, this behavior can be influenced by setting different weights. Especially the change in the controllability should be mentioned. Compared to the value at the initial point we obtain a significant improvement in the value of *control*.

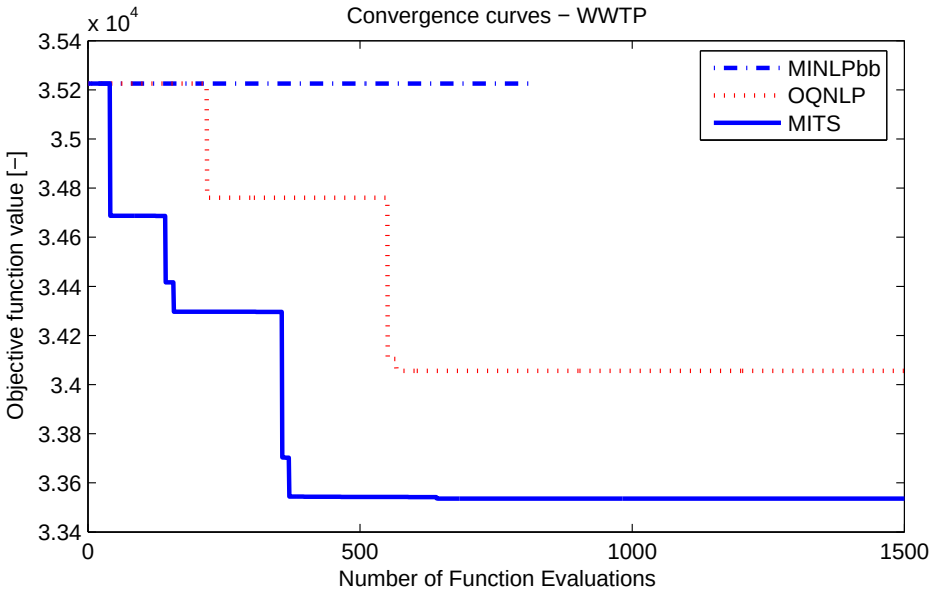

Figure 2: WWTP - Convergence curves

The performance of MITS is compared with the performance of two other solvers, namely, OQNLP and MINLPbb. OQNLP is known as one of the best global optimization solvers for black box optimization (see a comparison by Neumaier et al. [18]). The second solver MINLPbb was developed by Leyffer [15] and uses branch and bound to solve the MINLP. Figure 2 shows the convergence curves of the three different solvers. MITS and OQNLP stopped after the time limit of four days exceeded. MINLPbb is a deterministic approach and stopped after the internal criterion was satisfied. In both categories - objective function value and convergence speed - MITS outperforms the other solvers.

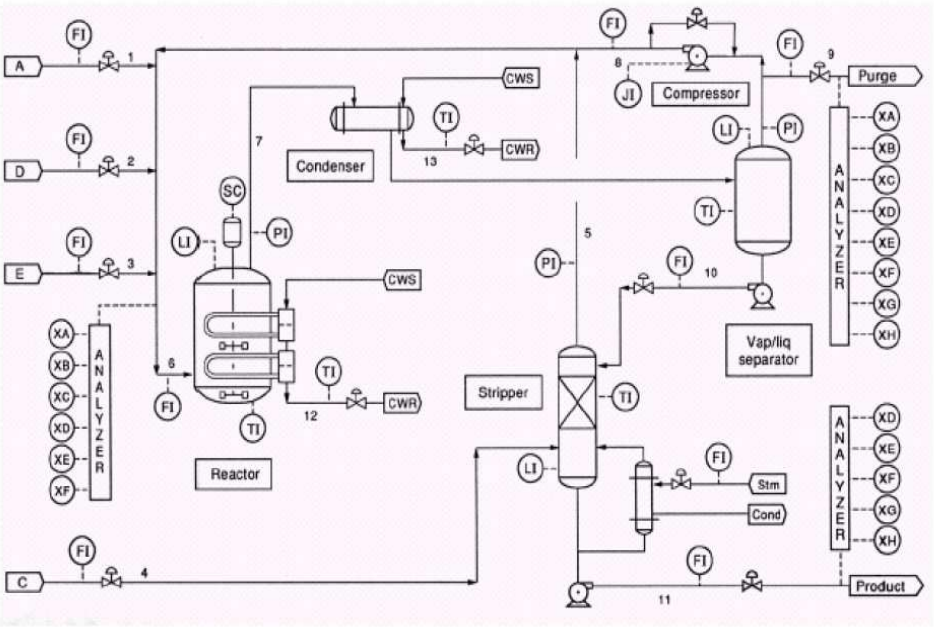

Figure 3: The Tennessee Eastman Process Flowsheet

4.2 The Tennessee Eastman Process

Since the publication of the Tennessee Eastman Process (TEP) example by Downs and Vogel [9], it has been widely used in the literature as a benchmark due to its challenging properties from a control engineering point of view: it is highly nonlinear, open-loop unstable and it presents a large number of measured and manipulated variables which offer a wide set of candidates for possible control strategies. The flowsheet for the TEP is depeicted in Figure 3. Two products (G and H) are produced from four reactants (A, C, D and E). A further inert trace component (B) and one byproduct (F) are present. The process units consists of a continuous stirred tank reactor, a condenser, a flash drum and a stripper. The gaseous reactants are fed to the reactor where they are transformed into liquid products. The following reactions take place in gas phase: These reactions are irreversible and exothermic with rates that depend on temperature through Arrhenius expressions and on the reactor gas phase concentration of the reactants. The reaction heat is removed from the reactor by a cooling bundle. The products and the unreacted feeds pass through a cooler and, once condensed, they enter a vapour-liquid separator. The noncondensed components recycle back to the reactor feed and the condensed ones go to a product stripper in order to remove the remaining reactants by stripping with feed stream. Products G and H are obtained in bottoms. The inert (B) and the byproduct

(F) are mainly purged from the system as a vapour from the vapour-liquid separator.

$$\begin{aligned} A_{(g)} + C_{(g)} + D_{(g)} &\rightarrow G_{(l)} \\ A_{(g)} + C_{(g)} + E_{(g)} &\rightarrow H_{(l)} \\ A_{(g)} + E_{(g)} &\rightarrow H_{(l)} \\ 3D_{(g)} &\rightarrow 2F_{(l)} \end{aligned} \tag{26}$$

Recently, Antelo et al. [3] applied their systematic approach to a plant-wide control design developed in a previous work (Antelo et al. [2]) to derive robust decentralized controllers for the Tennessee Eastman Process. In this framework, the TEP is represented as a process network. Then, conceptual mass and energy inventory control loops for each node are designed first to guarantee that the states of the plant will remain on a convex invariant region, where the system will be passive and therefore input-output stability can be stated (Antelo et al. [2]). The next step is to realize the proposed conceptual inventory control loops using the physical inputs-outputs of the process. Some extra control loops are needed to achieve the convergence of the intensive variables since the inventory control by itself does not ensure the convergence of these variables to a desired operation point. In some cases, the available degrees of freedom are not enough to implement the complete control structure that ensures both extensive and intensive variables convergence to the reference values. As a consequence, the setpoints of the inventory controllers can be used as new manipulated variables to complete the decentralized control design.

We explain the alternatives we introduced to extend the original hierarchical control design proposed by Antelo et al. [3]. Concerning the reactor level control loop in the original design, its set point modifies the reference for the flow controller acting over *E feed*. As an alternative for closing this loop, *D Feed* is proposed as manipulated variable.

For the reactor pressure case, the original proposal by Antelo et al.(2006) considers the condenser cooling water flow as the manipulated variable. By using this, the reactor pressure can be varied since the separator pressure and, as a consequence, the recycle rate can be modified. It is at this point where the control over the vapor mass inventory in the separator by using the purge rate is established in order to ensure that all inventories in the TEP will remain bounded and, therefore, input-output stability is guaranteed (Antelo et al., 2006).

As an alternative, we are considering here a manipulated variable widely used in the literature for the reactor pressure control loop: the purge flow. By modifying this, it is possible to regulate the separator pressure as well as the recycle flow, and therefore the reactor pressure. When this alternative is considered, an extra loop controlling the separator temperature (energy inventory) by acting over the condenser coolant flow is defined. Note that we are not considering other alternatives to control the reactor pressure as the A Feed, that is a disturbance in the model, or C Feed.

In order to determine the best control alternative among the proposed ones, a new binary vector $\mathbf{b}$ is added to our system dynamics. These 0-1 variables express which of

the four control strategies is being used, and they are defined as follows:

$$\begin{array}{ll} b_1 \in \{0,1\} & (E\ feed) \\ b_2 \in \{0,1\} & (D\ feed) \\ b_3 \in \{0,1\} & (Condenser\ coolant) \\ b_4 \in \{0,1\} & (Purge\ flow) \end{array} \tag{27}$$

Therefore, the original control design proposal by Antelo et al. (2006) will be characterized by the vector $b = (1,0,1,0)^T$ since it uses E Feed to control the reactor level and the condenser coolant flow to control the reactor pressure.

From all the exposed, the optimization problem consists now on solving the following MINLP of the form:

$$\begin{array}{l} \min\limits_{v,b} J(x,v,b) \\ \\ s.t. \\ f(\dot{x},x,p,v,b,t) = 0 \\ h(x,p,v,b) = 0 \\ g(x,p,v,b) \geq 0 \\ b_1 + b_2 = 1 \\ b_3 + b_4 - 1 \geq 0 \\ v^l \leq v \leq v^u \\ b^l \leq b \leq b^u \end{array} \tag{28}$$

where $b \in \{0,1\}^4$ is the vector of binary variables (0-1 variables) and $v \in \mathbb{R}^{36}$ are the continuous variables (the controller parameters). The lower and upper bounds for the binary variables will be of the form $b^l = (0,...,0)^T$ and $b^u = (1,...,1)^T$. It must be pointed out that we are considering that only one of the two alternatives for each loop can be active at one time, being necessary to introduce the additional linear constraints $b_1 + b_2 = 1$. The linear constraint $b_3 + b_4 - 1 \geq 0$ ensure that at least one of the alternatives b_3 or b_4 is active. The rest of the decision variables are connected to the tuning of the PI controllers.

The objective function proposed by Downs and Vogel (1993) in the TEP definitions is based on the operating costs and can be defined as follows:

$$\begin{array}{c} TC = PC \cdot PR + PrC \cdot PrR + CC \cdot CW + SC \cdot SR \\ \\ TC = \ 7.5973 \frac{\$}{kmol} \cdot PC + 0.1434 \frac{\$}{kmol} \cdot PrR + 0.0536 \frac{\$}{kWh} \cdot CW + 0.0318 \frac{\$}{kg} \cdot SC \end{array} \tag{29}$$

where TC are the total operating costs at the base case, PC and PR are the purge costs and purge flowrate, respectively. Analogously, PrC, CC and SC are the costs associated to the product stream, compressor and steam, and PrR, CW and SR are the product rate, the compressor work and the steam rate, respectively. Operating costs for this process are primarily determined by the loss of raw materials (in the purge, in the product stream and by means of the two side reactions). Economic costs for the process

are determined by summing the costs of the raw materials and the products leaving in the purge stream and the product stream, and using an assigned cost to the amount of F formed. The costs concerning the compressor work and the steam to the stripper are also included. Note that the objective function used in the MINLP formulation will be the mean of these operating costs along the whole simulation time horizon. For this work, this simulation time horizon was set to $t = 10\ h$. This is enough time for stabilization of the TEP.

After all these considerations concerning the objective function, the problem can be represented as an MINLP of the form (28):

$$v \in \mathbb{R}^{36},\ b \in \{0,1\}^4 : \quad \begin{array}{l} \min \overline{J(x,v,b)} = \overline{Total\ operating\ costs\ at\ base\ case} \\ v_0 - 0.5v_0 \leq v \leq v_0 + 0.5v_0, \end{array} \tag{30}$$

The lower and upper bound for the decision variables have been set to be the $\pm 50\%$ of the initial value for the decision vector. The reason for this selection is to avoid as much as possible the saturation problems that can be exhibited by the valves. These situations have been detected in preliminary dynamic simulations when considering a value of $\pm 100\%$ of v_0 as bounds for the decision vector.

Note that the NLP is also made up of the following constraints which are related with the reactor pressure, temperature and volume, and with the separator and the stripper volumes.

$$\begin{array}{c} P_{reactor} \leq 3000\ Kpa \\ 2\ m^3 \leq V_{reactor} \leq 24\ m^3 \\ T_{reactor} \leq 175\ ^{\circ}C \\ 1\ m^3 \leq V_{separator} \leq 12\ m^3 \\ 1\ m^3 \leq V_{stripper} \leq 6\ m^3 \end{array} \tag{31}$$

Note that changes in the decision variables (v) will be translated into variations in the states x that can even drive the system to shutdown due to the fact that one or more of the constraints defined in (31) have been violated. More precisely, the reactor volume, pressure and temperature are related with the pairs of decisions variables (gain-time constant) $v(11)$-$v(30)$, $v(12)$-$v(31)$ and $v(16)$-$v(35)$, respectively. Finally, the separator and the stripper volumes are linked with the pairs $v(9)$-$v(28)$ and $v(10)$-$v(29)$, respectively.

In order to solve the MINLP (28) problem we used the solvers MITS and OQNLP. The dynamic model for the proposed thermodynamic-based control design has been developed also as a SIMULINK code in the Process Engineering Group of the IIM-CSIC. Both solver started at the same initial point. As stopping criterion we set the maximum of function

evaluations equal to 10000. Table 4 lists the initial point and the best point located by MITS.

Table 4: Result for the TEP

Decision Variable	Control Loop	Default Values	Result MITS
$v(1)$	*A Feed Flow*	0.001	$1.4029e^{-3}$
$v(2)$	*D Feed Flow*	0.003	0.004043
$v(3)$	*E Feed Flow*	$1.8e^{-10}$	$9.6882e^{-11}$
$v(4)$	*C Feed Flow*	20	30.0
$v(5)$	*Condenser Coolant*	$7e^{-7}$	$9.8858e^{-7}$
$v(6)$	*Separator Flow*	$4e^{-4}$	$4.5727e^{-4}$
$v(7)$	*Stripper Flow*	0.004	0.004242
$v(8)$	*Production Rate*	3.2	4.8
$v(9)$	*Stripper Level*	-0.02	-0.021864
$v(10)$	*Separator Level*	-0.05	-0.048376
$v(11)$	*Reactor Level*	10	8.5024
$v(12)$	*Reactor Pressure*	-0.0001	$-5e^{-5}$
$v(13)$	*%G in Product*	-0.032	-0.02811
$v(14)$	*%A in Purge*	0.0009	5.664^{-4}
$v(15)$	*Recycle Rate*	0.00125	0.001271
$v(16)$	*Reactor Temp.*	-8	-8.1636
$v(17)$	*Separator Temp.*	100	90.219
$v(18)$	*G/H Product Ratio*	32	21.762
$v(19)$	*G/H Product Ratio*	46	30.891
$v(20)$	*A Feed Flow*	$1.6667e^{-5}$	$1.584e^{-5}$
$v(21)$	*D Feed Flow*	$1.6667e^{-5}$	$1.4535e^{-5}$
$v(22)$	*E Feed Flow*	4.1667	3.1266
$v(23)$	*C Feed Flow*	0.1667	0.83334
$v(24)$	*Condenser Coolant*	4.1667	2.0834
$v(25)$	*Separator Flow*	$1.6667e^{-5}$	$1.42116e^{-5}$
$v(26)$	*Stripper Flow*	$1.6667e^{-5}$	$1.302e^{-5}$
$v(27)$	*Production Rate*	2	1.2995
$v(28)$	*Stripper Level*	0.3333	0.19213
$v(29)$	*Separator Level*	3.3333	1.6704
$v(30)$	*Reactor Level*	0.01667	0.011874
$v(31)$	*Reactor Pressure*	0.3333	0.18142
$v(32)$	*%G in Product*	1.6667	1.0927
$v(33)$	*%A in Purge*	9.3667	10.284
$v(34)$	*Recycle Rate*	25	30.338
$v(35)$	*Reactor Temp.*	0.125	0.06475
$v(36)$	*Separator Temp*	8.3333	6.4701

$b(1)$	*Reactor Level (E Feed)*	1	0
$b(2)$	*Reactor Level (D Feed)*	0	1
$b(3)$	*Reactor Press. (Cond. Coolant)*	1	0
$b(4)$	*Reactor Press. (Purge Rate)*	0	1
Cost Value ($/h)		156.843	84.289

Note that for the case of the binary variables in Table 4, the manipulated variable that has been chosen to close the loop is presented in brackets. The solution obtained by MITS for the binary vector is $b = (0, 1, 0, 1)^T$. This vector defines the new realization of the control loops for the pressure and the level in the reactor by acting over the purge and the D Feed, respectively. Figure 4 shows the convergence curves for both solvers. MITS outperforms OQNLP in both, the objective function value and the convergence speed.

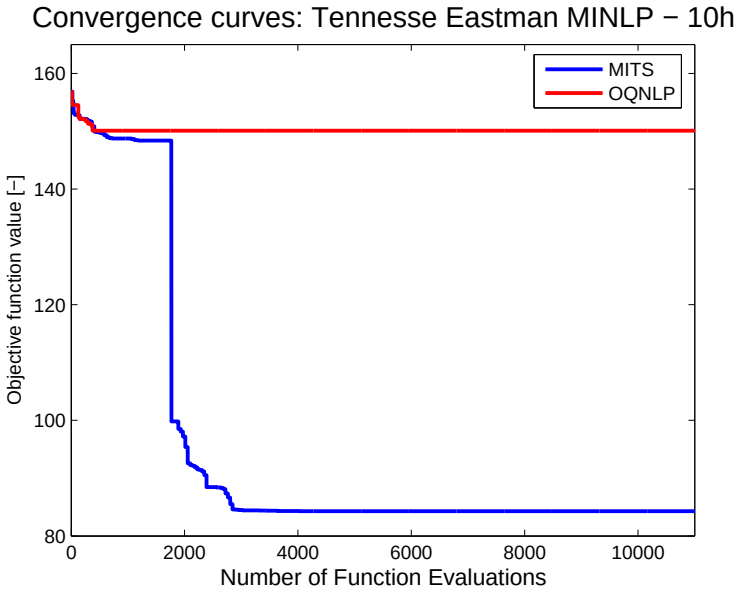

Figure 4: TEP - Convergence curves

5 Conclusions

We developed a hybrid algorithm for the optimization of the integrated process and control system design problem. The problem formulation as a mixed integer nonlinear program (MINLP) was presented. This hybrid strategy, MITS, uses a combinatorial component, based on Tabu Search, to guide the search into promising areas, and a local solver, which is activated to precisely approximate local minima. A Matlab implementation of this technique was developed, and its performance and robustness was tested on a two benchmark problems: a wastewater treatment plant for nitrogen removal and the Tennessee Eastman Process. MITS showed very good results, outperforming several selected MINLP solvers.

References

[1] Alex J., Bteau J.F., Copp J.B., Hellinga C., Jeppsson U., Marsili-Libelli S., Pons M.N., Spanjers H. and Vanhooren H. (1999):*Benchmark for evaluating control strategies in wastewater treatment plants*, Proc. ECC'99 Conference, Karlsruhe, Germany, August 31 -September 3, 1999.

[2] Antelo L. T., Otero-Muras I., Banga J.R. and Alonso A. A. (2005): *A systematic approach to plant-wide control based on thermodynamics*, Submitted to Computers & Chemical Engineering. Presented in the 15th European Symposium on Computer Aided Chemical Engineering (Proceedings of ESCAPE-15), 2005, 20(B), 1105-1110.

[3] Antelo L. T., Otero-Muras I., Banga J.R. and Alonso A.A. (2006): *Hierarchical Design of Decentralized Control Structures for the Tennessee Eastman Process*, Submitted to Industrial and Chemistry Engineering Research.

[4] Banga J.R., Moles C.G. and Alonso A.A. (2003): *Global optimization of bioprocesses using stochastic and hybrid methods*, In: Frontiers In Global Optimization, C.A. Floudas and P. M. Pardalos, (Eds.), Nonconvex Optimization and Its Applications, vol. 74, pp. 45-70, Kluwer Academic Publishers, ISBN 1-4020-7699-1.

[5] Bansal V. (2000): *Analysis, design and control optimization of process systems under uncertainty*, PhD dissertation, Imperial College of Science, Technology and Medicine, London.

[6] Battiti R. and Tecchiolli G. (1996): *The continuous tabu search: Blending combinatorial optimization and stochastic search for global optimization*, Annals of Operations Research, 63, 153-188.

[7] Copp J. (2002): *The COST simulation benchmark: Description and simulator manual*, Office for official publications of the European Community, Luxembourg, 154.

[8] COST 624: *Optimal Management of Wastewater Systems*, Available: http://www.ensic.inpl-nancy.fr/COSTWWTP/

[9] Downs J.J. and Vogel E.F. (1993): *A plant-wide industrial process control problem*, Computers & Chemical Engineering, 17, 245-255.

[10] Esposito W. R. and Floudas C.A. (2000): *Deterministic Global Optimization in Nonlinear Optimal Control Problems*, J. Global Optim. 17, 97-126.

[11] Exler O. and Schittkowski K. (2006): *A trust region SQP algorithm for mixed-integer nonlinear programming*, Optimization Letters, DOI 10.1007/s11590-006-0026-1.

[12] Glover F. and Laguna M. (1997): *Tabu search*, Kluwer Academic Publishers.

[13] Henze M., Grady Jr. C.P.L., Gujer W., Marais G.V.R. and Matsuo T. (1986): *Activated Sludge Model no. 1*, Technical Report n 1, IAWQ, London, Great Brittain.

[14] Jeppsson U. and Pons M.N. (2004): *The COST Benchmark simulation model - current state and future perspective*, Control Engineering Practice 12 (Editorial), 299-304.

[15] Leyffer S. (2001): *Integrating SQP and branch-and-bound for mixed integer nonlinear programming*, Computational Optimization and Applications, 18(3), 259-309.

[16] Moles C.G., Gutierrez G, Alonso A.A. and Banga J.R. (2003): *Integrated Process Design and Control Via Global Optimization: A Wastewater Treatment Plant Case Study*, Chemical Engineering Research and Design 81, no. 5, 507-517.

[17] Morari M. and Perkins J.D. (1994): *Design for Operations*, In Foundations of Computer-Aided Process Design (FOCAPD).

[18] Neumaier A., Shcherbina O., Huyer W. and Vinko T. (2005): *A comparison of complete global optimization solvers*, Math. Program., Ser. B 103, 335-356.

[19] Papamichail I. and Adjiman C. S. (2002): *A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations*, Journal of Global Optimization 24(1), 1-33.

[20] Pistikopoulos E.N., Ross R. et al. (1999): *Towards the Integration of Process Design, Process Control and Process Operability. Current Status and Future Trends*, Foundations of Computer-Aided Process Design (FOCAPD). Snowmass, Colorado.

[21] Sakizlis V., Perkins J.D. and Pistikopoulos E.N. (2004): *Recent advances in optimization-based simultaneous process and control design*, Computers & Chemical Engineering 28, 2069-2086.

[22] Schweiger C.A. and Floudas C.A. (1997): *Interaction of Design and Control: Optimization with Dynamic Models*, In: Optimal Control: Theory, Algorithms, and Applications. Hager W.W. and Pardalos P.M. (Eds.), Kluwer Academic Publishers, 1997.

[23] Sendn O.H., Moles C.G., Alonso A.A. and Banga J.R. (2004): *Multi-Objective Integrated Design and Control using Stochastic Global Optimization Methods*, In CACE Book on "Integration of Design and Control", M. Georgiados and P. Seferlis (Eds.), pp. 555-581, Elsevier Science, ISBN: 0-444-51557-7.

[24] Singer A. B., Bok J. K. and Barton P.I. (2001): *Convex Underestimators for Variational and Optimal Control Problems*, Computer Aided Chemical Engineering 9, 767-772.

[25] Spickenreuther T. (2005): *Entwicklung eines allgemeinen Branch & Bound Ansatzes zur gemischt-ganzzahligen Optimierung*, Diploma thesis, Department of Mathematics, University of Bayreuth.

[26] Takács I., Patry G.G. and Nolasco D. (1991): *A dynamic model of the clarification-thickening process*, Water Research, vol. 25(10), 1263-1271.

[27] Yuan Y.-X (1995): *On the convergence of a new trust region algorithm*, Numerische Mathematik 70, 515-539.