



CHALMERS

Chalmers Publication Library

Sheet-metal press line parameter tuning using a combined DIRECT and Nelder-Mead algorithm

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

IEEE 16th Conference on Emerging Technologies and Factory Automation, ETFA 2011, Toulouse, 5-9 September 2011

Citation for the published paper:

Svensson, B. ; Nia, N. ; Danielsson, F. (2011) "Sheet-metal press line parameter tuning using a combined DIRECT and Nelder-Mead algorithm". IEEE 16th Conference on Emerging Technologies and Factory Automation, ETFA 2011, Toulouse, 5-9 September 2011

<http://dx.doi.org/10.1109/ETFA.2011.6059031>

Downloaded from: <http://publications.lib.chalmers.se/publication/150588>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

Sheet-Metal Press Line Parameter Tuning using a Combined DIRECT and Nelder-Mead Algorithm

Bo Svensson¹, Nima K. Nia², Fredrik Danielsson¹, and Bengt Lennartson³

¹Dep of Engineering Science
University West
SE-461 86 Trollhättan, Sweden
bo.svensson@hv.se
fredrik.danielsson@hv.se

²Volvo Cars IT
Volvo Cars
SE- 293 80 Olofström, Sweden
NNIA@volvocars.com

³Dep of Signals and System
Chalmers University of
Technology
SE-412 96 Göteborg, Sweden
bengt.lennartson@chalmers.se

Abstract

It is a great challenge to obtain an efficient algorithm for global optimisation of nonlinear, nonconvex and high dimensional objective functions. This paper shows how the combination of DIRECT and Nelder-Mead algorithms can improve the efficiency in the parameter tuning of a sheet-metal press line. A combined optimisation algorithm is proposed that determines and utilises all local optimal points from DIRECT algorithm as Nelder-Mead starting points. To reduce the total optimisation time, all Nelder-Mead optimisations can be executed in parallel. Additionally, a Collision Inspection Method is implemented in the simulation model to reduce the evaluation time. Altogether, this results in an industrially useful parameter tuning method. Improvements of an increased production rate of 7% and 40% smoother robot motions have been achieved.

1. Introduction

A sheet-metal press line is a complex industrial automated manufacturing system that is manoeuvred by tuning of control parameters, e.g. times, positions, velocities and paths. It is a very complicated and time consuming task to find the control parameter set which gives the best production performance, i.e. the global optimum. Furthermore, due to changed circumstances, e.g. increased demands, increased cost, and less labours, the desirable production performance is from time to time modified, in form of different and/or combined targets, e.g. high production rate, soft motions, low wear, and low energy consumption.

The complexity in parameter tuning is due to the highly non-linear objective functions with multiple local optima and the considerable number of tuning parameters. In the test case presented in this work the number of control parameters is reduced to ten in each of totally five press stations. Hence, due to the multiple local optima, there exist several parameter sets that will result in increased production performance of the press line, without reaching the global optima. The challenge

is then; to determine a suitable optimisation algorithm that finds a control parameter set which, within a time limit constraint, improves the production performance as much as possible.

In complex applications, such as sheet-metal press lines, it is hard to find one single algorithm that, within reasonable time, handles both the global search and the local convergence. In order to tune a sheet-metal press line, Svensson et al. [1] have implemented and evaluated the two different optimisation algorithms, DIRECT [2] and Nelder-Mead [3]. Both algorithms presented acceptable results, but each with its disadvantage. The DIRECT algorithm has a very slow convergence, i.e. it requires many evaluations for acceptable results. The Nelder-Mead algorithm is highly dependent on the choice of starting point, i.e. in reality a “good guess” based on knowledge of where to start searching. One approach to face the challenge to determine a suitable algorithm is to combine two algorithms with different properties [4]. In this work a new combination of DIRECT and Nelder-Mead is suggested and evaluated for a sheet metal press line.

Due to the fact that an optimisation algorithm requires a considerable number of evaluations, the time of each evaluation is essential. To reduce the evaluation time, a new Collision Inspection Method, CIM [5], is introduced to reach a reasonable total optimisation time. CIM is designed to in advance identify possible collision points that could occur in the simulation. Then, in each evaluation, a simplified and fast collision detection algorithm is executed.

2. Sheet-metal press line

In this work the sheet-metal press line 53 at Volvo Cars in Gothenburg, Sweden, has been used as a case study, see Figure 1. This actual press line has five press stations and in-between these there are four combined feeders/extractors (two axis robots) and four intermediate stations for moving and rotating components both vertically and horizontally (three axis manipulators). An additional feeder/extractor (two axis

robot) and a conveyor are located before the first press station, and after the final press station. Furthermore, there are numerous digital and analogue sensors, actuators, push-buttons, indicators, etc. in the press line. Five identical PLCs work together and control the entire press line, handling discrete events, continuous feedbacks, motions, supervisory and safety control. There is in total more than 2000 in- and outputs, 90 control parameters, 25 servo axis, and 10 robot paths in this automated sheet-metal press line. And, in addition, more than 100 000 lines of control code in the PLCs.

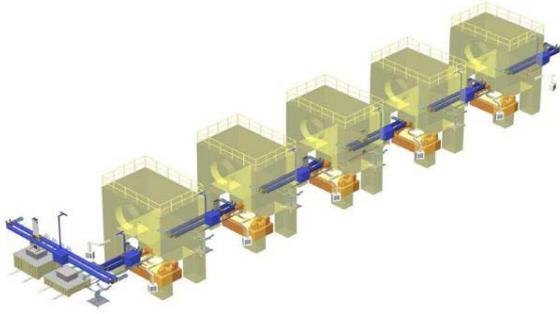


Figure 1. Sheet-metal press line.

Today the press line is mainly optimised by tuning of one robot path and ten control parameters in each of the five press stations. The remaining paths and parameters are also of importance but have minor influence in the final tuning. However, the parameter tuning is a complicated task, due to five individual control systems with interlocking signals in between. Even if these control systems are individual, they are strongly dependent on each other. Furthermore, the complexity increases significantly when considering that the extractor in one press station is physically the same robot as the feeder in the next press station.

To be able to optimise such a complex automated manufacturing system a model was created in a previous work [6]. The model is based on a synchronised hardware-in-the-loop, SHIL, method. A successful implementation and validation of the sheet-metal press line 53 SHIL model is also presented in [6]. In this work the SHIL model is used without any further development as the response function.

3. Press line parameter tuning method

The off-line parameter tuning method described in Svensson [7] has been used in this work, see Figure 2. To summarise, all sheet-metal press line control parameters p_j that are to be tuned, e.g. times, positions, velocities, paths, etc. are gathered in a parameter vector

$$\mathbf{p} = [p_1 \quad p_2 \quad \dots \quad p_n] \quad (1)$$

and subject to the constraint $\mathbf{p} \in Q_p \subseteq \mathbf{R}^n$ consisting of tuning process parameter limitations of the form

$$p_{j,\min} \leq p_j \leq p_{j,\max} \quad (2)$$

where $p_{j,\max}$ and $p_{j,\min}$ are upper and lower parameter limits for each separate parameter p_j to be tuned. The number of press line control parameters used in this work is $n=10$.

The process optimiser initialises and starts one evaluation of the sheet-metal press line model in the SHIL simulation, with a defined initial parameter vector $\mathbf{p}^{(1)}$. The evaluation in the SHIL simulation yields the response $h(\mathbf{p})$. From received $h(\mathbf{p})$ the process optimiser calculates individual production performance values $g_i = g_i(h(\mathbf{p}))$. Based on the given objective function f the optimisation algorithm then calculates the next parameter vector $\mathbf{p}^{(2)}$ to be evaluated in the press line model. This iterative procedure is then repeated over again until a given number of evaluations m is reached and the resulting optimally tuned process parameters in \mathbf{p}^* are established.

The objective function f has to be varied in order to reach different and/or combined industrial targets. The press line objective function constitutes an input to the process optimiser and is defined as

$$f(\mathbf{p}) = c_1 g_1(h(\mathbf{p})) + c_2 g_2(h(\mathbf{p})) + c_3 g_3(h(\mathbf{p})) \quad (3)$$

where $c_i \cdot 0$ are weight values to be chosen to represent a suitable combination of the production performances $g_i \cdot 0$. The same production performances are used in this work as in previous case study of comparison reasons, i.e.

g_1 = production rate, number of sheet-metals per time;
 g_2 = robot smooth motion, sheet-metal in gripper; and
 g_3 = robot smooth motion, empty gripper.

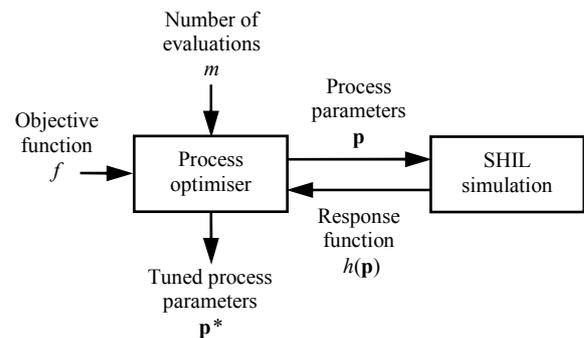


Figure 2. Off-line parameter tuning method.

4. Optimisation algorithms

Two different kinds of well-known and well-used global direct search optimisation algorithms are suggested, implemented and evaluated by Svensson et al. [1] in the press line parameter tuning method. The DIRECT algorithm as presented by Jones et al. [2] and

the simplex direct search algorithm, proposed by Nelder and Mead [3].

4.1. DIRECT algorithm

The DIRECT algorithm is designed in order to solve complex global optimisation problems using only objective function values [2]. The algorithm is a kind of Lipschitz optimisation, where the maximal derivative of the objective function is limited by a constant, discussed by Horst et al. [8]. In the DIRECT algorithm the need to specify this Lipschitz constant is however eliminated, Jones et al. introduce a different way of looking at the Lipschitz constant. One of the DIRECT algorithm's benefits is its needlessness of start values and its search in the entire parameters space, only limited by the parameters max and min.

The name DIRECT comes from the shortening of the phrase "DIviding RECTangles" and this is pretty indicative of how the algorithm works. Initially the DIRECT algorithm normalizes the entire search space into a unit hypercube, and determines the objective function at the centre point. In the first iteration the hypercube is divided into hyper-rectangles, i.e. trisected in every parameter direction, and all new centre points are evaluated. In subsequent iterations, the DIRECT algorithm balances the global and local searches by identifying potential optimal hyper-rectangles, based upon not only the value of the objective function at the centre but also the size of the hyper-rectangle. All potentially optimal hyper-rectangles are then divided into smaller hyper-rectangles and their centre points are evaluated. The DIRECT algorithm is guaranteed to converge to the global optimal function value, if the objective function is continuous or at least continuous in the neighbourhood of the global optimum.

The main weakness of the algorithm is the need of a very high number of evaluations. The DIRECT algorithm finds interesting areas rather quickly but has a very slow convergence close to the optimum points. This is a common behaviour of Lipschitz optimisation algorithms, as was pointed out by Hansen, et al. [9]. This weakness was also confirmed in previous implementation and evaluation of the DIRECT algorithm in the press line parameter tuning [1]. Another serious shortcoming of the DIRECT algorithm is the lack of an obvious stopping criterion, which also was evident in previous case study. Gablonsky [10] summarises the most frequently used DIRECT algorithm stopping criteria in the following order:

- a) number of iterations or function evaluations,
- b) function value close to the global minimum (only applicable if the global minimum is known), and
- c) size of the (smallest) hyper rectangle (several definitions exist).

However, the DIRECT algorithm is well-used as method for global optimisation problems, assuming short evaluation times. For instance, in the optimisation platform Tomlab [11], the global optimisation routines

`glbDirect`, `glbFast` and `glbSolve` are implementations of the DIRECT algorithm.

4.2. Nelder-Mead algorithm

The Nelder-Mead simplex algorithm is a classical widely accepted optimisation algorithm handling non-linear objective functions and making no use of the objective function derivatives [3]. Wright [12] stresses that one of the benefits of the Nelder-Mead algorithm is its low number of required evaluations, which is of importance when each evaluation is time consuming. In previous case study with the press line parameter tuning method, the Nelder-Mead algorithm proved its effectiveness on convergence, i.e. need of few evaluations. Another important benefit is its ease to implement and expand into many parameters.

A simplex is a geometrical figure consisting, in n dimensions, of $(n+1)$ points. If any point of a simplex is taken as the origin, the n other points define vector directions which span the n -dimension vector space. A simplex in two dimensions is a triangle; a simplex in three dimensions is a tetrahedron. Through a sequence of elementary geometric transformations (expansion, reflection, contraction, and multi-contraction), the initial simplex moves, expands or contracts. Expansion is enlarging the particular search area. By reflection, a new point located on the other side of the worst point against the centroid of the remaining points is generated. Contraction is to select the point halfway between the worst point and the centroid. Multi-contraction is the opposite of expansion, and is used when the optimum is surrounded by the simplex. For determining the appropriate transformation, the method uses only the values of the objective function at the considered points. After each transformation, the current worst point is replaced by a better one. This is repeated until the difference in the objective function values of the worst and best points is less than a specified tolerance; or the simplex has decreased to a single point. By this the simplex adapts itself to the objective function landscape and finally finds the optimum. At each step it is checked that the new generated point is not outside the allowed solution space.

The weakness of the Nelder-Mead algorithm is its need of a good starting point. There is a risk, dependent on the starting point, of the simplex 'crawling up' on a local optimum without continuing to the global one. Hence, it is not a real true global optimisation algorithm when the objective function is highly non-linear and has multiple local optima. The previous implementation and evaluation of the Nelder-Mead algorithm in the press line parameter tuning confirm this characteristic.

However, the Nelder-Mead simplex algorithm is a very well-known and well-used method, even if it lacks rigorous convergence results. For instance, the optimisation toolbox in Matlab uses Nelder-Mead as a global optimiser within the function `fminsearch`.

4.3. Need of combined algorithm

It is a great challenge to obtain one algorithm for global optimisation of nonlinear, nonconvex and high dimensional objective functions. To search for a global optimum, it is necessary to investigate the whole search domain. When a promising area is found, the optimum within that area must be found as quickly as possible. Both tasks are hardly performed through only one algorithm. To overcome that issue, several combinations of one global searching and one local converging optimisation algorithm have been proposed in the literature.

Through its excellent global behaviour the DIRECT algorithm offers itself as a good starting point generator for other, local optimisers, with fast convergence. Several different approaches, presented in the literature, have incorporated a local optimiser into DIRECT, either at the end of the optimisation or periodically during the DIRECT optimisation process, e.g. Cox et al. [13], Gablonsky [10], Jones [14], and Nelson and Papalombros [15].

The Nelder-Mead algorithm is a well-tuned local optimisation algorithm with fast convergence. Hence, it is often used as a local optimiser in combination with other algorithms. Either periodically during the global optimisation, or utilising generated starting point from completed global search. Several examples are presented in the literature, e.g. Chelouah and Siarry [16], [17], Durand and Alliot [18], and Yang and Douglas [19]. However, to the author's knowledge, no DIRECT and Nelder-Mead algorithm combination is described in the literature.

5. Combined optimisation algorithm

Based on the literature and previous case study results, discussed in Section 4, a combination of the DIRECT and the Nelder-Mead algorithms is proposed to obtain an efficient press line parameter tuning algorithm. Due to DIRECT's excellent global search qualities it forms an ideal starting point generator. At the end of the global optimisation the Nelder-Mead algorithm, proven to be simple and efficient, is utilised as a competent local optimiser. As switch-over condition the number of evaluations is selected. This selection is based on the fact of being the most common way to stop DIRECT, if not the global optimum value is known [10], and the industrial requirement of a limited, known total optimisation time. Moreover, not only the best found point from DIRECT initiates a local optimisation, as it is in many combined algorithms, e.g. [10] and [16]. The proposed optimisation algorithm, as shown in Figure 3, determines instead all local optima in the set of DIRECT evaluated points. Each of these local optima constitutes starting points in separate Nelder-Mead algorithms, possible to evaluate in parallel in order to reduce the total optimisation time. The proposed algorithm consists

of four steps, namely; global search for starting points, selection of starting points, parallel local optimisation and finally determination of optimally tuned process parameters.

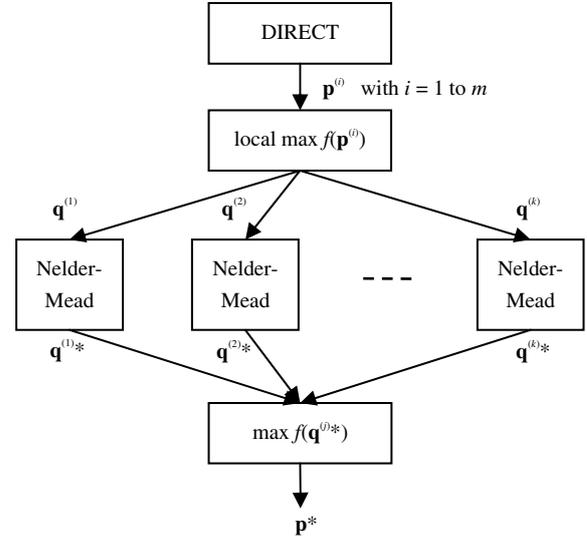


Figure 3. Proposed combined optimisation algorithm.

5.1. Global search for starting points

The combined optimisation algorithm starts searching globally with the DIRECT algorithm as a starting point generator. The DIRECT algorithm iterates until the predefined number of evaluations, m , are reached. The outcome set

$$P = \{ \mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(m)} \} \quad (4)$$

is the set of all evaluated points (parameter vectors) \mathbf{p} in the DIRECT algorithm, with corresponding objective function values $f(\mathbf{p})$.

5.2. Selection of starting points

A. Divide P in lower and upper sets for each parameter

Gather all points in P whose parameter value is less than the value of the same parameter in point $\mathbf{p}^{(i)}$ in the lower set

$$P_{j,-}^{(i)} = \{ \mathbf{p} \in P \mid p_j < p_j^{(i)} \} \quad (5)$$

where $p_j^{(i)}$ denotes parameter j in $\mathbf{p}^{(i)}$. Correspondingly the upper set

$$P_{j,+}^{(i)} = \{ \mathbf{p} \in P \mid p_j > p_j^{(i)} \} \quad (6)$$

is defined. Note that if $p_j^{(i)}$ is equal or close to its lower or upper parameter limits, $p_{j,\min}$ or $p_{j,\max}$, the associated set might be empty.

B. Determine neighbours for each parameter

The lower neighbour to point $\mathbf{p}^{(i)}$ regarding parameter j is the point in the lower set with shortest Euclidean distance to $\mathbf{p}^{(i)}$, i.e.

$$\mathbf{p}_{j,-}^{(i)} = \arg \min_{\mathbf{p} \in P_{j,-}^{(i)}} \sqrt{\sum_{j=1}^n (p_j^{(i)} - p_j)^2} . \quad (7)$$

Correspondingly the upper neighbour to point $\mathbf{p}^{(i)}$ regarding parameter j is defined

$$\mathbf{p}_{j,+}^{(i)} = \arg \min_{\mathbf{p} \in P_{j,+}^{(i)}} \sqrt{\sum_{j=1}^n (p_j^{(i)} - p_j)^2} . \quad (8)$$

Observe that a lower, or upper, neighbour does not exist if the associated set is empty. Together all lower and upper neighbours for all parameters form the set

$$P^{(i)} = \bigcup_{j=1}^n \{ \mathbf{p}_{j,-}^{(i)}, \mathbf{p}_{j,+}^{(i)} \} \quad (9)$$

which constitutes the set of neighbours of point $\mathbf{p}^{(i)}$ to be used next to figure out if $\mathbf{p}^{(i)}$ appears to be a local optimum or not.

C. Determine local optima

A local optimum is defined as a point that has an objective function value, defined as in (3), greater or equal to its neighbours, defined in (9). All local optima in the set P , of evaluated points from the DIRECT algorithm, form the subset

$$Q = \{ \mathbf{p}^{(i)} \in P \mid f(\mathbf{p}^{(i)}) \geq f(\mathbf{p}), \forall \mathbf{p} \in P^{(i)} \} . \quad (10)$$

The set Q will at least include one point, the one with best found objective function value. The number of local optima is unpredictable and depends on the characteristic of the sheet-metal press line objective function.

5.3. Parallel local optimisation

Each local optimal point \mathbf{q} in the set

$$Q = \{ \mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \dots, \mathbf{q}^{(k)} \} \quad (11)$$

constitutes a starting point for the Nelder-Mead algorithm. Each point in Q can of course be evaluated in parallel. The outcome from each local optimisation becomes a locally tuned point \mathbf{q}^* .

5.4. Determination of tuned process parameters

All locally tuned points \mathbf{q}^* from the k Nelder-Mead local optimisation processes are gathered in the set

$$Q^* = \{ \mathbf{q}^{(1)*}, \mathbf{q}^{(2)*}, \dots, \mathbf{q}^{(k)*} \} \quad (12)$$

forming a final set of possible candidates to \mathbf{p}^* . The objective function values of all locally tuned points are

compared, and the best point determines the optimally tuned parameter vector as

$$\mathbf{p}^* = \arg \max_{\mathbf{q} \in Q^*} f(\mathbf{q}) . \quad (13)$$

Note that the predefined number of evaluations, m , constitutes the stopping criterion for the DIRECT part of the combined optimisation algorithm. However, the Nelder-Mead stopping criterion is based on convergence, implying that the obtained number of Nelder-Mead evaluations may vary from trial to trial. Hence, the total number of evaluations is in forecast only an estimated value.

6. Collision inspection method

Even with the proposed new combined optimisation algorithm a considerable number of evaluations are necessary. Consequently the time for each single evaluation in the SHIL simulation is of importance. With the sheet-metal press line model implemented as in previous case study [1] one single evaluation requires on average 8 minutes. Hence, decreasing the evaluation time is obviously desired. The Collision Inspection Method, CIM, proposed by Nia et al. [5] constitutes an effective candidate to decrease the evaluation time. CIM is a collision detection method designed to identify possible collision points that could occur in a simulation in advance. CIM eliminates the need of an extensive and time consuming collision detection tool, e.g. RobCad in previous case study, and applies a simplified collision detection algorithm. CIM is divided in two parts; one pre-processing part, which is only calculated once for each type of component in the sheet-metal press line; and one executing part, which is executed in every evaluation. CIM is dependent on a general path/direction, and all solid geometry of interest must be sampled. Dies, press, handling equipment and sheet-metal parts are all complex 3D objects and the transport of a sheet is done along a known path. These attributes of press lines makes them appropriate for simplification with CIM.

The pre-processing part of CIM handles detecting collision between objects of interest. Collision points are stored in a chosen Tool Centre Point, TCP, and are compressed to a collision curve. As an example the movable object in Figure 4 has its TCP on the top. By moving the object in sampling steps of dY and dZ , collision points are detected on the surface of the fixed object. At the same time as a collision point is detected, the position of TCP in space is stored. The stored TCP points result in a collision curve. Thus with the consideration of small variations in the third dimension: Given CAD models for the movable object, the fixed object and minimum sampling distances $dXYZ$, a collision detection algorithm returns a vector collision curve including the position (xyz coordinates) of a

chosen TCP when collision occurs between the movable and fixed objects. In cases where the simulation demands a change in the geometry of the tested CIM objects, new collision curves must be calculated with CIM.

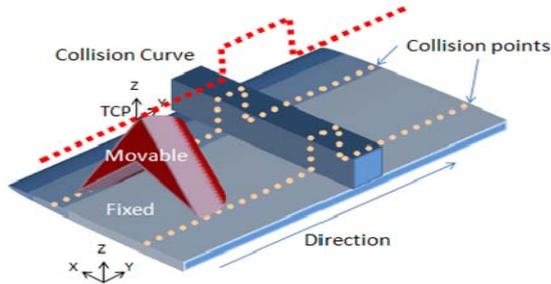


Figure 4. Illustration of the pre-processing part of the Collision Inspection Method.

The calculation algorithm for CIM can be implemented in any program which handles CAD data. In this work the algorithm was implemented in Catia V5. The time for calculating the collision curves in the sheet-metal press line 53 was around 1.5 hours. This time is dependent on the performance of the computer, and the collision algorithm used in the program; hence there are still improvement potentials in the creation of the collision curve.

The benefit of using CIM is in the executing part of the method. It detects collisions between the movable object and the fixed object by an algorithm which, for every evaluation, checks if the TCP is above the collision curve in the collision free area. This part of CIM uses minimal computational resources since complex 3D objects have been reduced to a TCP and a collision curve. Consequently, a huge reduction in evaluation time will be seen.

7. Results

All off-line parameter tuning results in this work are expressed as percentage change in corresponding value from the on-line tuned sheet-metal press line. It is vital to notice that the manual on-line parameter tuning has been accomplished by a skilled operator. A negative sign on a result implies that the off-line parameter tuning has not succeeded to reach a better value than the on-line tuning.

A parameter tuning only focused on production rate, i.e. with weight values $c_1=1.0$, $c_2=0$, and $c_3=0$ in the sheet-metal press line objective function, has been performed. The production rate was increased 7 percentages but with side-effect of a more jerky motion (i.e. non-smooth), see Table 1. These non-smooth motions may lead to increased wear of the equipment, increased energy consumption and even critical fall out of the sheet-metal component, which are unacceptable in real production. Due to that, the rest of the experiments

were performed with a weighted combination of production rate and smooth motions in the objective function.

Table 1. Percentage change in individual components of the sheet-metal press line objective function.

	Production rate	Smooth motion, plate in gripper	Smooth motion, empty gripper
Only prod rate	7.0	-71.6	-79.0
Experiment 1	6.5	39.5	1.5
Experiment 2	4.7	37.7	41.8
Experiment 3	4.3	38.7	41.0

Table 2. Combinations of weight values in sheet-metal press line objective function in respectively experiment.

	c_1	c_2	c_3
Experiment 1	1.0	0.1	0.1
Experiment 2	1.0	0.3	0.2
Experiment 3	1.0	0.4	0.2

How to select the production performance weight values to reach different and/or combined targets, e.g. high production rate, soft motions, low wear, and low energy consumption, is an industry related economic task. This needs further investigation and is not evaluated in this work. However, to illustrate some industrial optimisation demands, three different combinations of the weight values c_1 , c_2 , and c_3 in the sheet-metal press line objective function have been set up as Experiment 1, 2 and 3, see Table 2. Experiment 1 fulfils a requirement of soft robot motions when handling sheet-metal components, combined with a high production rate. Experiment 2 corresponds to a focus on decreased wear of the production equipment, but still with an acceptable production rate. Experiment 3 is aimed at soft robot motions with emphasis on motions with sheet-metal components in the gripper. Achieved results in individual components of the sheet-metal press line objective function are summarised in Table 1.

The DIRECT algorithm finds interesting areas rather quickly but has a very slow convergence close to the optimal points. Hence, the number of evaluations, m , used as stopping criterion for the DIRECT algorithm is vital. In the three experiments, with the combined optimisation algorithm, three different values of m has been selected, 800, 1200 and 1400 evaluations. Results from these nine trials, in form of percentage change in press line objective function, are shown in Table 3. In the table is also shown, as comparison, the corresponding results from DIRECT as a solitary algorithm with the stopping criterion 10 000 evaluations.

The results, in Table 3, illustrate that the combined algorithm has a distinct advantage in achieving an improved objective function value. Even with $m=800$ evaluations the objective function value is close to the best found. If the number of evaluations in the combined algorithm is extended, a better objective function value

is reached than DIRECT algorithm solitary does not find even in 10 000 evaluations. In two experiments the best found value is reached already after $m=1200$ evaluations, in the third a slightly further increase appear for $m=1400$. If this further increase of 0.33 percentages is worth the time of 200 additional evaluations is an industry related economic question, not evaluated in this work.

Table 3. Percentage change in objective function, combined optimisation algorithm compared with DIRECT solitary.

	Combined algorithm $m = 800$	Combined algorithm $m = 1200$	Combined algorithm $m = 1400$	DIRECT 10 000 evaluations
Experiment 1	9.0	9.5	9.5	9.4
Experiment 2	17.3	17.7	18.0	17.4
Experiment 3	21.5	21.8	21.8	21.3

However, the total number of evaluations in the combined algorithm is not only m , the local Nelder-Mead optimisation also require evaluations. The Nelder-Mead algorithm stopping criterion is simplex decreased to a single point, and accordingly number of evaluations is not predictable. The number of required Nelder-Mead evaluations in all performed trials is presented in Table 4. Based on performed trials, the average number of evaluations for the local Nelder-Mead optimisation is 154, with a variation between 113 and 224. Still, the total number of evaluations will not exceed $m+one$ Nelder-Mead evaluations, since all local Nelder-Mead optimisations can be executed in parallel.

The combined algorithms advantage of performing local optimising of all local optima, not only the best found point, is obvious in the results shown in Table 4. In two trials it is a local optimum, not equal to the best found point from DIRECT evaluations, which start the Nelder-Mead that reaches the optimal tuned point \mathbf{p}^* . Furthermore, in two other trials there is two best found points and only one of them lead to \mathbf{p}^* in each case. Only in five of nine trials there is a single best found point that reaches the optimal tuned point \mathbf{p}^* in the ending local search. This illustrates the importance of running multiple Nelder-Mead optimisation processes in the proposed combined algorithm.

The introduced combined optimisation algorithm reduces the number of required evaluations. In addition also a reduction of the time for each evaluation is performed. The collision detection tool, RobCad, in the sheet-metal press line SHIL model from previous work [1], has been replaced with CIM. With this exchange a reduction from 8 minutes to, in average, 30 seconds per evaluation has been reached. Thus, an efficient and useful parameter tuning method for sheet-metal press lines is achieved. The case study tuning experiments, with all Nelder-Mead local optimisations in parallel and $m=800$ have an average optimisation time of 8 hours; $m=1200$ a time of 11 hours; and $m=1400$ a time of 13 hours. These optimisation times depends mainly on the

evaluation times in the SHIL simulation. The time contribution of the selection and booth optimisation algorithms described in Section 5 is close to zero.

Table 4. Percentage change in objective function, before and after local Nelder-Mead optimisation of local optima.

	Before Nelder-Mead	After Nelder-Mead		Nelder-Mead evaluations
Experiment 1 $m = 800$	5.82	8.96	\mathbf{p}^*	166
	5.37	7.47		136
	-9.51	7.04		208
Experiment 1 $m = 1200$	6.57	7.79		122
	6.57	9.54	\mathbf{p}^*	173
	6.41	8.43		161
	-7.80	8.32		224
Experiment 1 $m = 1400$	6.81	7.92		129
	6.81	8.22		150
	6.57	7.79		122
	6.57	9.54	\mathbf{p}^*	173
	6.41	9.17		171
	-7.80	8.32		224
Experiment 2 $m = 800$	16.17	17.33	\mathbf{p}^*	157
	16.15	16.48		133
	-15.04	2.62		186
Experiment 2 $m = 1200$	16.17	17.33		157
	16.17	17.67	\mathbf{p}^*	143
	16.15	16.48		133
	15.91	17.11		118
	-15.04	2.62		186
Experiment 2 $m = 1400$	17.03	18.00	\mathbf{p}^*	130
	16.65	17.51		129
	16.57	17.37		141
	16.17	17.33		157
	16.17	17.67		143
	16.15	16.48		133
	-15.04	2.62		186
Experiment 3 $m = 800$	20.00	21.51	\mathbf{p}^*	158
	-15.49	15.93		184
Experiment 3 $m = 1200$	20.31	21.75	\mathbf{p}^*	138
	20.21	21.31		116
	20.00	21.20		136
	20.00	21.51		158
	19.97	20.40		113
	-15.49	15.93		184
Experiment 3 $m = 1400$	20.75	21.24		141
	20.31	21.75	\mathbf{p}^*	138
	20.00	21.20		136
	20.00	21.51		158
	19.97	20.40		113
	-15.49	15.93		184

8. Conclusions

The proposed combination of DIRECT and Nelder-Mead optimisation algorithms has, in the performed case study, shown to improve the efficiency in parameter tuning of a sheet-metal press line. There is a great challenge to obtain an optimisation algorithm that handles highly non-linear objective functions with multiple local optima and considerable number of tuning parameters, as it is in sheet-metal press lines. Within a

time limit constraint the proposed optimisation algorithm has shown to determine a control parameter set which improve sheet-metal press line performances, both in terms of 7 percentages increased production rate and 40 percentages smoother robot motions.

Compared to earlier presented combinations, found in literature, the proposed new algorithm has a distinct advantage. Not only best found point from the DIRECT algorithm, but also all local optima, are used as starting points for the Nelder-Mead local optimisation. Only in five of nine trials, in the case study, the single best found point initiate the local optimiser that reaches the optimal tuned process parameters in \mathbf{p}^* .

The total optimisation time is reduced by parallel execution of local Nelder-Mead optimisations. To further decrease the optimisation time, the Collision Inspection Method, CIM, was implemented in the model. A pre-processing part of CIM executed only once, handles all complex calculations of the collision detection. This work confirms that an efficient and useful sheet-metal press line parameter tuning method is achieved when the proposed combined optimisation algorithm and CIM is implemented.

Acknowledgement

The authors thank Damien Ringenbach, Stephane Torres and Pierre-Henri Biau, students from ENSIL engineering school, University of Limoges, France, for their excellent contribution in this work.

References

- [1] B. Svensson, F. Danielsson, and B. Lennartson, "Simulation Based Optimization of a Sheet-Metal Press Line" *Proc. of 14th IEEE International Conference on Emergency Technologies and Factory Automation*, Mallorca, Spain, pp. 447-454, 2009.
- [2] D.R. Jones, C.D. Perttunen, & B.E. Stuckman, "Lipschitzian optimization without the Lipschitz constant" *Journal of Optimization Theory and Applications*, Vol: 79, Iss: 1, pp. 157-181, 1993.
- [3] J.A. Nelder, and R. Mead, "A Simplex Method for Function Minimization" *Computer Journal*, Vol: 7, pp. 308-313, 1965.
- [4] J.D. Pinter, *Global Optimization in Action*, Kluwer Academic Publishers, Dordrecht/Boston/London, 1996.
- [5] N.K. Nia, F. Danielsson, and B. Lennartson, "A faster collision detection method applied on a sheet metal press line" *Proc. of 21st International Conf on Flexible Automation and Intelligent Manufacturing*, Taiwan, 2011.
- [6] B. Svensson, F. Danielsson, and B. Lennartson, "A Virtual Real-Time Model for Control Software Development - applied on a Sheet-Metal Press Line" *Proc. of 3rd International Industrial Simulation Conference*, Berlin, Germany, pp. 119-123, 2005.
- [7] B. Svensson, "Optimisation of Manufacturing Systems Using Time Synchronised Simulation" *Lic Thesis*. Chalmers University of Technology, Gothenburg, Sweden, 2010. Available at: <http://publications.lib.chalmers.se/records/fulltext/126326.pdf>
- [8] R. Horst, P. M. Pardalos, and N.V. Thoai, "Introduction to Global Optimization" *Kluwer Academic Publishers*, Dordrecht, The Netherlands, 2nd ed, 2000.
- [9] P. Hansen, B. Jaumard, and S.H. Lu, "Global Optimization of univariate Lipschitz functions: II" *New Algorithms and Computational Comparison*. Mathematical Programming, 55:273-292, 1992.
- [10] J.M. Gablonsky "Modifications of the Direct Algorithm" *PhD Thesis*. North Carolina State University, Raleigh, North Carolina, 2001.
- [11] Tomlab Optimization Environment. Available at: <http://tomopt.com/tomlab> [Accessed at Apr 15, 2011].
- [12] M.H. Wright, "Direct search methods: once scorned, now respectable" *Proc. of 1995 Dundee Biennial Conference on Numerical Analysis*, pp. 191-208, Addison-Wesley, Reading, MA. 1995.
- [13] S.E. Cox, R.T. Haftka, C.A. Baker, B. Grossman, W.H. Mason, and L.T. Watson, "A comparison of global optimization methods for the design of a high-speed civil transport" *Journal of Global Optimization*, Vol: 21, pp. 415-433, 2001.
- [14] D.R. Jones, "DIRECT global optimization algorithm" *Encyclopedia of optimization* (eds. C.A. Floudas and P.M. Pardalos), pp. 431-440. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
- [15] S.A. Nelson, and P.Y. Papalambros, "A modification to Jones global optimization algorithm for fast local convergence" *Proc. of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp 341-348, St. Louis, MO, 1998.
- [16] R. Chelouah, and P. Siarry, "Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multim minima functions" *European Journal of Operational Research*, Vol: 148, Iss: 2, pp 335-348, 2003.
- [17] R. Chelouah, and P. Siarry, "A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multim minima functions" *European Journal of Operational Research*, 161, pp. 636-654, 2005.
- [18] N. Durand, and J.M. Alliot, "A combined Nelder-Mead simplex and genetic algorithm" *Proc. of the Genetic and Evolutionary Computation Conference GECCO99*, pp. 1-7, Orlando, FL, USA, 1999.
- [19] R. Yang, and I. Douglas, "Simple genetic algorithm with local turning-efficient global optimizing technique" *Journal of Optimization Theory and Applications*, 98(2), pp. 449-465, 1998.