Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

Multi-objective ant colony optimization for the twin-screw configuration problem

Cristina Teixeira [a], J. A. Covas [a], Thomas Stützle [b] & A. Gaspar-Cunha [a]

[a] IPC/I3N-Institute for Polymer and Composites, University of Minho, Guimarães, Portugal

[b] Université Libre de Bruxelles, IRIDIA, Brussels, Belgium

Available online: 27 Feb 2012

To cite this article: Cristina Teixeira, J. A. Covas, Thomas Stützle & A. Gaspar-Cunha (2012): Multi-objective ant colony optimization for the twin-screw configuration problem, Engineering Optimization, 44:3, 351-371

To link to this article: http://dx.doi.org/10.1080/0305215X.2011.639370

Engineering Optimization
Vol. 44, No. 3, March 2012, 351–371

Multi-objective ant colony optimization for the twin-screw configuration problem

Cristina Teixeira[a], J.A. Covas[a], Thomas Stützle[b] and A. Gaspar-Cunha[a]*

[a]*IPC/I3N-Institute for Polymer and Composites, University of Minho, Guimarães, Portugal;* [b]*Université Libre de Bruxelles, IRIDIA, Brussels, Belgium*

(*Received 28 January 2011; final version received 18 October 2011*)

The twin-screw configuration problem (TSCP) consists of identifying the best location of a set of available screw elements along a screw shaft. Owing to its combinatorial nature, it can be seen as a sequencing problem. In addition, different conflicting objectives may have to be considered when defining a screw configuration and, thus, it is usually tackled as a multi-objective optimization problem. In this research, a multi-objective ant colony optimization (MOACO) algorithm was adapted to deal with the TSCP. The influence of different parameters of the MOACO algorithm was studied and its performance was compared with that of a previously proposed multi-objective evolutionary algorithm and a two-phase local search algorithm. The experimental results showed that MOACO algorithms have a significant potential for solving the TSCP.

Keywords: multiple objectives; ant colony optimization; evolutionary algorithms, polymer extrusion; twin-screw configuration

1. Introduction

In virtually all polymer processing and compounding technologies, the raw material (pellets or powder) is melted and subsequently given the desired shape. Feeding and solids conveying, melting, mixing and melt pumping are performed in a special section of the processing equipment known as plasticating unit, which usually consists of one or two (counter- or co-rotating) Archimedes-type screws rotating inside a heated barrel. In turn, the plasticating unit is coupled to, for example, a die or a mould, in the case of extrusion and injection moulding, respectively. This article deals with optimization problems that arise when setting the configuration of co-rotating twin-screw extruders. These machines are progressively finding more applications in polymer compounding (manufacture of masterbatches, compounds, polymer blends, composites and nanocomposites), powder coating, food processing (cereals, biscuits, cookies, crackers, pet food) and hot melt extrusion of pharmaceuticals (drug delivery, transmucosal and transdermal systems, *etc.*) (White *et al.* 2010, Sakai 1991). They combine continuous production, easy operation,

*Corresponding author. Email: agc@dep.uminho.pt

ISSN 0305-215X print/ISSN 1029-0273 online

http://dx.doi.org/10.1080/0305215X.2011.639370
http://www.tandfonline.com

modular construction and good yield ranging from a few hundred grams per hour in laboratory machines to 100 tonnes per hour in the biggest industrial machines.

In co-rotating twin-screw extruders, two parallel screws that are mechanically intermeshed rotate in the same direction and at the same speed inside a hollow heated barrel. Frequently, each screw is assembled by sliding along a shaft a certain number of individual elements that were selected from a larger pool. Extruder manufacturers offer a wide variety of geometries, with distinct conveying and mixing characteristics (for an overview of available screw elements, see Sakai 1991 and Kohlgrüber 2007). In general, in laboratory-scale extruders, the barrel can also be constructed by coupling individual segments.

The modular construction provides the possibility of adapting the machine features to the specifications of different productions. This applies not only to the type and sequence of unit operations to be performed (such as solids conveying, melting, feeding of a solid additive, injection of a liquid, mixing, devolatilization, pumping and shaping), but also to the characteristics of some of these operations. For example, mixing can be made to range from purely distributive to mostly dispersive, *i.e.* from a mere spatial rearrangement of the material ingredients to the rupture of existing agglomerates into primary particles (Kohlgrüber 2007).

The performance of a manufacturing process using twin-screw extruders is obviously determined by the thermophysical and rheological characteristics of the raw materials (melt viscosity, melting temperature, thermal stability, density, composition, *etc.*), the set operating conditions (temperature profile, screw speed and feed rate of the various components) and the screw (and barrel) geometrical profile. Thus, assembling a screw from individual screw elements is an essential but challenging task, which might dictate *a priori* the success of the entire process. The twin-screw configuration problem (TSCP) consists of defining the position of a set of available screw elements along a screw shaft such that it maximizes the performance of a specific application. Currently, this task is performed empirically on a trial-and-error basis, making use of available process knowledge. Commercial process modelling software is seldom used to decrease the number of experimental iterations (Vergnes *et al.* 1998, White *et al.* 2001). As an alternative, an automatic approach to twin-screw design formulating it as a combinatorial sequencing problem, which is solved using an optimization procedure coupled to a modelling routine, has been proposed (Gaspar-Cunha *et al.* 2002, 2005). The method employs a multi-objective evolutionary algorithm (MOEA) (Deb 2001, Coello Coello *et al.* 2007), denoted as reduced Pareto set genetic algorithm (RPSGA), that uses a clustering technique to simultaneously ensure the progression of the population of solutions in the direction of the Pareto-optimal front and the dispersion of the solutions along this front (Gaspar-Cunha *et al.* 1997, Gaspar-Cunha and Covas 2004, Gaspar-Cunha 2000). The approach was successfully applied to the optimization of operating conditions (Gaspar-Cunha *et al.* 2002), screw configuration (Gaspar-Cunha *et al.* 2005) and, more recently, material manufacture via reactive extrusion (starch cationization (Teixeira *et al.* 2010, Gaspar-Cunha *et al.* 2011) and ε-caprolactame copolymerization (Gaspar-Cunha *et al.* 2005, Teixeira *et al.* 2010). The authors are unaware of other attempts to develop a scientific methodology to solve the TSCP. Still, Potente and Thumen (2006) addressed the problem of optimizing the geometry of individual screw elements (but not locating them spatially in the shaft) making use of an aggregation of quality functions.

Modelling adequately the flow and heat transfer (and, eventually, the chemical conversion and morphological evolution) of a material along the axis of a twin-screw extruder is rather complex, involving numerical solutions that are computationally demanding. Consequently, the evaluation of each screw configuration takes a significant computation time and any optimization run becomes slow owing to the inherent intensive use of the modelling routine. Thus, it is of practical interest to develop and/or implement as efficient as possible optimization algorithms. A first step attempted by the authors consisted of applying multi-objective extensions of local search algorithms to the TSCP. In particular, the use of iterative improvement algorithms embedded into the two-phase local search (TPLS) framework provided good results in comparison with RPSGA (Teixeira *et al.* 2011).

Ant colony optimization (ACO) is a relatively recent and powerful metaheuristic for tackling combinatorial problems. ACO takes inspiration from real ants' foraging behaviour to define algorithmic solutions to computationally hard optimization problems. The first ACO algorithm (Ant System) was proposed by Dorigo *et al.* in the early 1990s. Later followed the definition of the ACO metaheuristic framework by Dorigo and Di Caro (1999), which provides a high-level model for ACO algorithms. In ACO algorithms, ants exchange information indirectly by depositing pheromone that influences the choice of other ants. For a recent overview on ACO algorithms and their applications refer to Dorigo and Stützle (2010).

Several approaches have been proposed to apply ACO algorithms to multi-objective optimization problems (MOOP) such as multi-objective scheduling, vehicle routing and portfolio selection (Mariano and Morales 1999, Iredi *et al.* 2001, Doerner *et al.* 2004, García-Martinez *et al.* 2007, Angus and Woodward 2009, López-Ibáñez and Stützle 2009, 2010). A detailed review and comparison of the available multi-objective ACO (MOACO) algorithms for solving the bi-objective travelling salesman problem can be found in García-Martinez *et al.* (2007). A more recent review provides a detailed classification of MOACO algorithms (Angus and Woodward 2009). Several design alternatives of how to extend ACO algorithms to MOOPs have been studied (López-Ibáñez and Stützle 2009) and the algorithmic design choices in available MOACO algorithms have been analysed (López-Ibáñez and Stützle 2010).

This article investigates MOACO algorithms as an alternative solution method for the TSCP. In particular, different design choices for MOACO algorithms are considered and their impact on MOACO's performance is analysed for various bi-objective TSCP instances. Based on the insights gained from this study, the final MOACO algorithm for the TSCP is defined and its performance compared to that of RPSGA and a TPLS algorithm. The computational results show that the proposed algorithm results in approximations to the Pareto front that are better than those of RPSGA on several TSCP instances. This indicates that MOACO algorithms are very promising for tackling the TSCP.

The article is structured as follows. Section 2 presents the characteristics of the TSCP as well as the relevant aspects of the modelling routine. The multi-objective algorithms studied are described in Section 3 and the examples to be analysed are presented in Section 4. Section 5 contains a discussion of the MOACO parameters and a comparison between MOACO, MOEA and TPLS results. Finally, the main conclusions are given in Section 6.

2. Twin-screw configuration problem

2.1. *The extrusion process*

Figure 1 illustrates the typical layout of a co-rotating twin-screw extruder. The machine contains two parallel identical screws that can rotate at constant (but tuneable) speed inside a heated barrel. The material enters upstream via a gravimetric or volumetric feeder set to work at a given rate assuring that the screws will work partially filled along most of their length. By action of the screw rotation, the material progresses axially with a complex flow pattern (it is repeatedly transferred between the channels of the two screws), while it is subjected to distinct thermomechanical environments (shear stresses, temperature, residence time), depending on the local screw geometrical features.

As it progresses downstream, the material melts owing to the combined effect of heat conduction from the barrel, heat generation due to friction and plastic deformation. Further downstream the melt may cross mixing zones that induce variable levels of distributive and/or dispersive mixing,

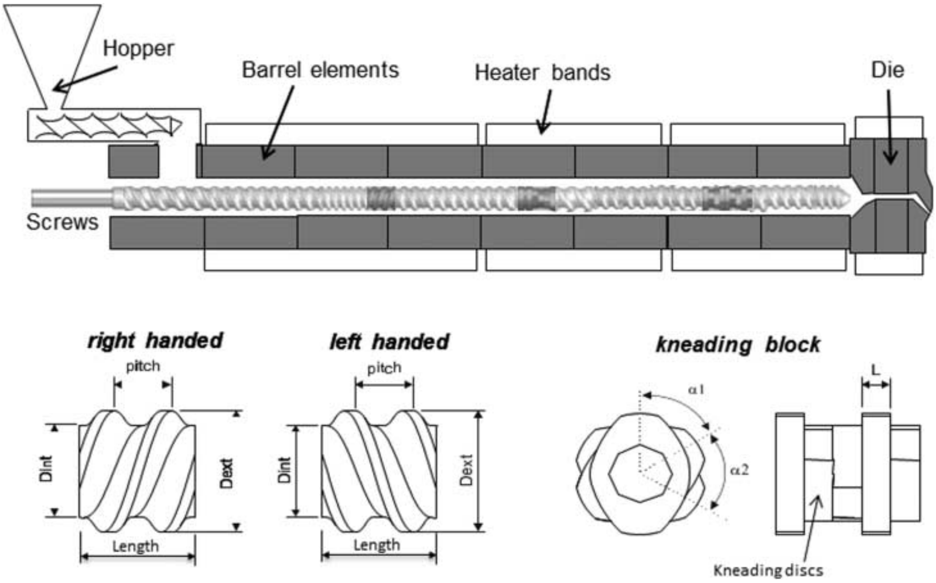

Figure 1. Co-rotating twin-screw extruder. Top: example of machine layout; bottom: types of screw elements.

flow along partially filled channels where new components—typically fillers or reinforcements—may be added, or devolatilization is attempted. A repetition of these operations along the screw is also possible. The sequence will depend on screw and barrel design, location and type of accessories and operating conditions. Towards the screw tips, the melt is pressurized and flows through the shaping die (White *et al.* 2001, White and Kim 2010, Vergnes *et al.* 1998, Gaspar-Cunha *et al.* 2002). Screw speed and output can be controlled independently, although they both affect flow and heat transfer inside the machine.

As explained above, in most machines the screws are built by assembling a number of screw elements. These can vary in type (see Figure 1) and, for each, variations in pitch, helix or staggering angle, and length or number of kneading discs exist. Right-handed elements have a positive helix angle, which gives them good conveying capacity (the higher the angle, the larger the output capacity); thus, here the screws will work partially filled (this saves energy, allows for temperature relaxation and is useful for secondary feeding or devolatilization). Conversely, the negative angle of left-handed screw elements imposes a restriction to the flow, *i.e.* they generate pressure and may cause filling of a few screw channels upstream. Kneading blocks comprise a variable number of kneading discs, which can be staggered at positive, neutral (*i.e.* 90°), or negative angles (in practice, the helix angle can also vary within the same kneading block). In the first case, they behave as conveying sections (but with a good distributive mixing capacity), while in the third case they become analogous to left-handed elements. The restrictive character increases the local residence time and hydrodynamic stress levels, thus inducing polymer melting (if they correspond to the first restrictive section upstream) and/or distributive and dispersive mixing (tuneable via the staggering angle). Therefore, the screw profile of the extruder depicted in Figure 1 comprises one melting/mixing and two mixing zones, separated by four conveying sections. As the inlet material progresses along the screw, it quickly approaches the first restrictive zone, where the combination of pressure, temperature and local residence time will cause melting. If this zone is long enough, substantial dispersive and distributive mixing will develop (and, when chemical reactions are also involved, high chemical conversions may be reached). Melt flow during the second and third mixing zones will essentially contribute to better dispersive mixing. In fact, this type of mixing requires exposure to sufficiently high stresses during sufficient time. However, it is preferable to design a screw with several mixing zones instead of having one of equivalent length, as pressure and viscous dissipation in the latter would reach prohibitive levels. On the other hand, this design will generate complex pressure profiles, as flow develops along sequences of partially filled and fully filled channels.

2.2. *Modelling routine*

Owing to the complexity of the flow and heat transfer in twin screw extruders and the need to generate sufficiently accurate predictions of the process response upon changing its main parameters, the authors developed a numerical modelling routine (Teixeira *et al.* 2007, 2010). For the purposes of this work, this routine can assumed to be a ‘black box’. The inputs are the material properties, the screw configuration (in the present case, the screw configuration is to be optimized) and the operating conditions (screw speed, barrel set temperature and mass output). The outputs are the parameters that characterize the process performance; these parameters are used as the objectives to be optimized.

It is clear that different screw configurations will create inherently different thermomechanical environments. The linkage between operating conditions, screw geometry/configuration, materials properties and the extruder performance is made through the modelling routine. Thus, the modelling routine must (1) be capable of describing accurately the relevant flow and heat transfer phenomena, (2) be sensitive to changes in geometrical or operational parameters, and (3) require moderate computational resources. A detailed description of the routine is given elsewhere (Teixeira *et al.* 2007, 2010). It encompasses all steps from material inlet to die exit, including solids conveying under or without pressure, melting, melt conveying under or without pressure and die flow. Figure 2 presents three different screw profiles that could be used in the extruder of Figure 1 and that are described in detail in Table 1. Each screw comprises 16 elements; right-handed and left-handed elements are identified by their length and pitch, while kneading

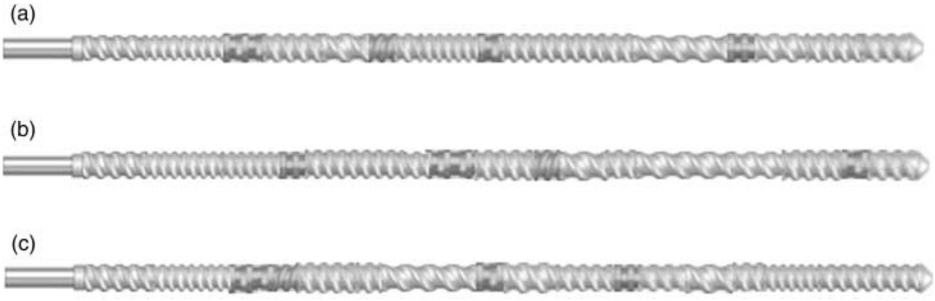

Figure 2. Examples of screw profiles to be used in the extruder of Figure 1.

Table 1. Geometrical identification of the screws illustrated in Figure 2.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	KB–45	30	60	–20	30	20	KB–60	30	30	60	KB–30	45	30	20
B	Length	97.5	120	60	30	120	30	45	60	30	30	30	120	60	60	37.5	30
	Pitch	45	30	20	KB–60	30	30	KB–45	30	–20	60	30	60	45	30	KB–30	20
C	Length	97.5	120	45	30	60	30	120	37.5	60	60	30	30	30	30	60	120
	Pitch	45	30	KB–45	–20	30	30	60	KB–30	45	30	KB–60	20	60	30	20	30

blocks (KB) are defined by their length and staggering angle. In industrial practice, the extrusion companies deal with at most 25–30 elements. Figure 3 and Table 2 demonstrate how the performance of these screws is distinct when using identical operating conditions (barrel and die set to 220 °C, screws rotating at 150 rpm and feed rate of 8 kg/h). A polypropylene polymer (ISPLEN PP 030G1E from Repsol) is being processed, having physical, thermal and rheological properties characterized by the manufacturer, or determined experimentally (see Domingues *et al.* 2010).

Figure 3 shows the evolution along the screw length of the average melt temperature, average degree of channel fill, cumulative mechanical power consumption, cumulative residence time, pressure and average shear rate. Table 2 contains the values of global process responses. Average strain is a measure of the extent of distributive mixing; specific mechanical energy (SME) represents the mechanical energy consumption per unit weight of processed material; viscous dissipation quantifies the increase in melt temperature relative to the local set value. Not only the values of these parameters change significantly with screw geometry, but they are also conflicting.

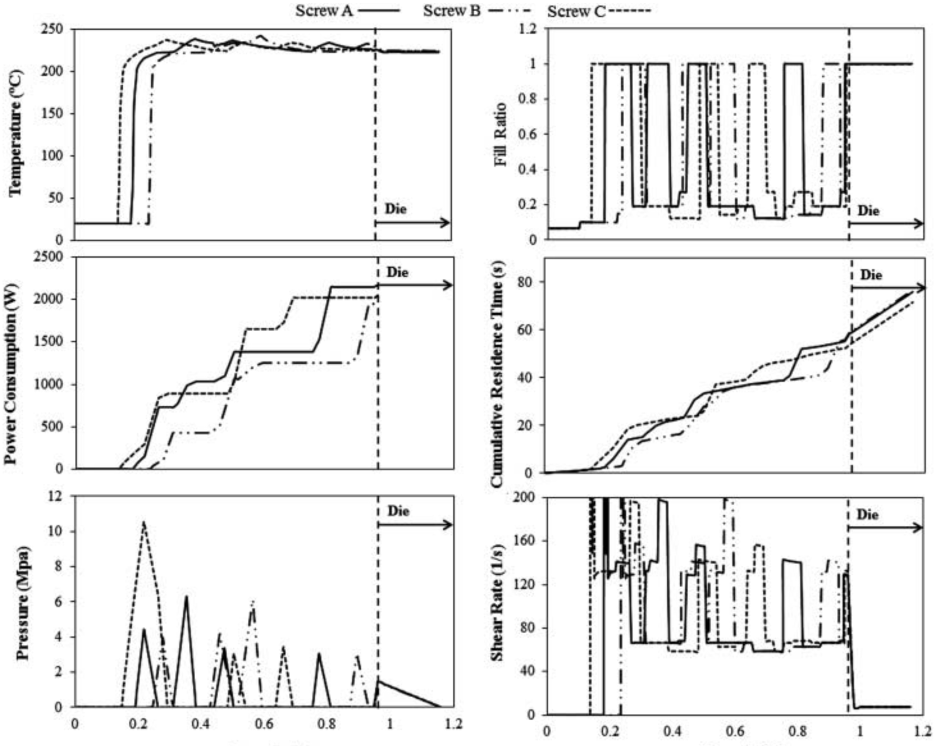

Figure 3. Evolution of various process parameters along the barrel, when using the screw profiles of Figure 2 under identical operating conditions.

Table 2. Global process responses when using the screw profiles of Figure 2 under identical operating conditions.

	Average strain	SME	Viscous dissipation
Screw A	7449.9	0.979	1.198
Screw B	7682.6	0.889	1.217
Screw C	6859.4	0.925	1.198

Note: SME = specific mechanical energy.

For example, the best distributive mixing is obtained for screw B (Table 2) at the cost of having to cope with higher viscous dissipation (which, if excessive, will cause premature material degradation). The performance measures presented in Table 2 will be adopted as objectives for the optimization runs discussed in subsequent sections.

2.3. *The problem to solve*

The examples presented above demonstrate the practical importance and difficulty in optimizing the process. In general, this should be done using as parameters to optimize (*i.e.* the decision variables) the operating conditions (Gaspar-Cunha *et al.* 2002), the screw configuration (Gaspar-Cunha *et al.* 2005), the individual geometry of the different screw elements, or a combination of these variables, depending on the practical situation to tackle. The operating conditions and individual geometry of the elements are continuous variables, while the screw configuration variables are discrete (since they define the location of the elements through the order in which the screw elements are put on the screw shaft). Simultaneously, the best screw configuration can be determined for a fixed set of screw elements previously selected, or by using a pool of elements available for the machine under consideration (Gaspar-Cunha *et al.* 2005). In the present work, the TSCP is dealt with using a fixed number of predetermined screw elements. Again, the TSCP consists of the definition of the best location along the screw axis of a number of distinct screw elements that optimizes the process performance. There are no restrictions in terms of the possible location of any screw element, except that the initial (upstream) elements must be of the conveying type (to ensure smooth material inlet) and at least one restrictive element (left-handed or kneading block) must be present in order to melt the material. Thus, the TSCP can be seen as a sequencing problem where the resources (screw elements) must be ordered sequentially along the screw shaft.

For the purposes of this work, an instance of the problem consists of a predetermined set of screw elements and their respective location along the screw shaft needs to be determined. For each instance, three different variants are defined by the particular combination of objectives to be considered. The objective values are computed by the modelling routine, *i.e.* and as seen above, for each screw configuration that is defined by the optimization algorithm, the modelling routine must be run. Each run of the modelling routine needs 2–3 central processing unit (CPU) minutes (depending mainly on the number of restrictive screw elements) on an AMD opteron TM 2116 dual-core processor running at 2.4 GHz with 2 MB L2 cache. Hence, the total computation time for one run of an optimization algorithm with a maximum of 3000 evaluations of sequences by the modelling routine is more than 6000 CPU minutes, that is, more than 4 CPU days. Clearly, an exhaustive search for all possible sequences is infeasible for even a small number of screw elements and therefore heuristic algorithms are required to determine good screw configurations.

3. Multi-objective optimization: concepts and algorithms

3.1. *Basic concepts*

An important aspect to take into consideration when applying an optimization algorithm to real-world problems is the representation of the solutions. This is particularly relevant in the present problem, since it constitutes an engineering problem where continuous and discrete variables can exist simultaneously. In the case of the TSCP, solutions are represented by discrete variables identifying the location of the screw elements. In particular, a solution can be represented as a permutation of the indices of the screw elements. This is also relevant for designing the different operators of the optimization algorithms, as will be illustrated below.

Most real-world optimization problems, including the TSCP, involve the simultaneous optimization of various criteria. The complexity of the task is increased by the existence of conflicting objectives (Deb 2001, Coello Coello *et al.* 2007). In MOOPs, solutions are evaluated with respect to an objective function vector $F(X) = (f_1(X), \ldots, f_N(X))$ that assigns to every solution $\mathrm{X} \in S$ a value for each objective function. Here, N is the number of objectives. The task is to find solutions for which the objective function vectors are in some sense optimal. Let us assume, without loss of generality, that maximization problems are considered. The dominance criterion defines a partial order on the set of solutions. In particular, for two given vectors $\boldsymbol{u} = (u_1, \ldots, u_\mathrm{N})$ and $\boldsymbol{v} = (v_1, \ldots, v_\mathrm{N})$, it is possible say that $\boldsymbol{u}$ dominates $\boldsymbol{v}$ in the Pareto sense, if and only if $\forall i \in \{1, \mathrm{K}, N\} : u_i \geq v_i$ and $\exists i \in \{1, \ldots, N\} : u_i > v_i$. A solution $X \in S$ is then said to be Pareto optimal with respect to S if and only if there exists no $X' \in S$ for which $\boldsymbol{v} = F(X') = (f_1(X'), \ldots, f_N(X'))$ dominates $\boldsymbol{u} = F(X) = (f_1(X), \ldots, f_N(X))$. If *a priori* it was not possible to make any assumption on the preferences of a decision maker, the goal is to obtain a set of feasible solutions that minimize F in the sense of Pareto optimality. This set of optimal solutions in the decision space is known as the Pareto-optimal set and the resulting representation in the objective space is named the Pareto front. MOEAs are an efficient method to solve MOOPs: taking inspiration from the natural evolution process, they use a population of solutions that can approximate the Pareto front (Deb 2001, Coello Coello *et al.* 2007). Different approaches for tackling MOOPs have been proposed in the literature (Deb 2001, Coello Coello *et al.* 2007).

3.2. *Multi-objective evolutionary algorithms*

The development of an evolutionary algorithm (EA) is strongly dependent on the characteristics of the problem to be solved (Goldberg 1989). The most important characteristics to be considered in the present work are the discrete nature of the decision variables and the presence of multiple conflicting objectives. Given the latter, the application of MOEAs seems to be straightforward. However, owing to the first characteristic of the problem, none of the MOEA algorithms available in the literature can be used without considerable modifications concerning solution representation and the evolutionary operators. In fact, only the global concepts of these algorithms can be transferred to the TSCP.

Therefore, a modified version of RPSGA, a MOEA that was developed for continuous problems by some of the authors of this article, was adopted here (Gaspar-Cunha *et al.* 1997, Gaspar-Cunha and Covas 2004, Gaspar-Cunha 2000). In fact, in earlier research, RPSGA was found to be competitive with NSGA-II (Deb *et al.* 2000), a standard MOEA proposed by Deb *et al.*, which justifies this choice. RPSGA is based on the use of two populations (internal and external), which evolve simultaneously during the successive generations. RPSGA uses a clustering technique to rank the solutions present in the population. This technique replaces the crowding distance operator of NSGA-II and improves the quality of the solutions found when dealing with more than two objectives.

Owing to the solution representation used for the TSCP (discrete variables that define a permutation of the screw element indices), the crossover and mutation operators in RPSGA were replaced by the inver-over operator (Tao and Michalewicz 1998). The main steps of the RPSGA algorithm are illustrated below (Algorithm 1).

ALGORITHM 1 *RPSGA for the TSCP*

```
(1) Random initial population (internal);
(2) Empty external population and archive;
(3) while Stopping-Condition not satisfied do
    (a) Evaluate internal population;
    (b) Calculate the Fitness of the individuals using clustering;
    (c) Copy the best individuals to the external population;
    (d) if the external population becomes full then
            Apply clustering to this population;
            Copy the best individuals to the internal population;
    end if
    (e) Select the individuals for reproduction;
    (f) Inver-over operator;
    (g) Copy the non-dominated solutions to the archive;
    (h) Discharge the dominated solutions from the archive;
end while
```

The algorithm starts with the creation of a random internal population of size M (step 1) and of external and archive populations of a maximal size $2M$, which are initially empty (step 2). The following steps are carried out in each iteration, until a stopping condition is satisfied. The solutions of the internal population are evaluated by the modelling routine (step 3a). Then, the fitness of each solution is calculated making use of the clustering technique (step 3b) and a fixed number of the best solutions (as defined by the clustering technique) are copied to the external population (step 3c). If the latter is not full, the selection and inver-over operators are applied in order to produce a new population (steps 3e and 3f). The non-dominated solutions found are copied to the archive (step 3g) and those that become dominated owing to the inclusion of new solutions are discharged from the archive (step 3h). If the external population becomes full, the clustering technique is applied to sort the individuals of the external population, and a predefined number of the best solutions (as defined by the clustering technique) are incorporated into the internal population by replacing the lowest fitness individuals (step 3d).

As said above, RPSGA uses a clustering technique that reduces the number of solutions on the efficient frontier while maintaining its characteristics intact; this enables the selection of the best solutions for reproduction and, simultaneously, it maintains the distribution of the solutions along the Pareto front (Gaspar-Cunha 2000). The clustering technique is applied by first dividing the set of solutions considered (either the internal or the external populations) by a predefined number of ranks (N_{ranks}). For the first rank (r equal to one), a clustering algorithm is applied until the number of solutions that rest is equal to (M/N_{ranks}), *i.e.* only this number of ranks exists. To these solutions, which represent the solutions that best represent the clusters, is attributed rank one (*i.e.* rank equal to r). For the second rank (r equal to two), the entire population is reduced to (r^*M/N_{ranks}). Those individuals that do not have yet a rank (attributed in the first iteration) are attributed the rank two. This process is repeated until the last rank. The non-dominated individuals are attributed the maximum rank value (*i.e.* N_{ranks}). A rank function is used to calculate a global evaluation function value and the individuals in the population are selected by a roulette wheel technique using this evaluation function value. In addition, an archive of non-dominated solutions is kept, to prevent good solutions being lost (Gaspar-Cunha 2000). For a more detailed description of RPSGA, refer to Gaspar-Cunha *et al.* (1997) Gaspar-Cunha and Covas (2004) and Gaspar-Cunha (2000).

3.3. *Multi-objective ant colony optimization*

ACO takes inspiration from the pheromone trail laying and following behaviour of real ants and transfers some core behaviours to an algorithmic approach for tackling complex combinatorial problems (Dorigo and Di Caro 1999). To apply ACO, pheromone trails are associated to solution components of the problem to be tackled. Artificial ants then iteratively generate solutions to the problem under concern using a probabilistic solution construction mechanism and update the pheromone trails based on a positive feedback loop. From a high level perspective, the outline of an ACO algorithm is as follows (Algorithm 2).

ALGORITHM 2 *Generic ACO algorithm*

```
(1) Initialize pheromone matrices to value $\tau_{\text{ini}}$;
(2) while termination conditions not meet do
    (a) Construct solutions;
    (b) Evaluate solutions;
    (c) Update pheromones;
    (d) Update archive;    % if ACO is applied to MOOPs in Pareto sense
end while
```

The ACO algorithm starts with the initialization of the pheromone matrix. While the termination conditions are not met (*i.e.* at each iteration) the following tasks are carried out. First, a solution is constructed for each ant, based on probabilistic values that are a function of the pheromone strengths associated to solution components and possibly available heuristic information (step 2a). Then, each solution is evaluated (step 2b). Next, the pheromones are updated (step 2c) by first reducing the pheromone trails by a fixed factor ρ (simulating pheromone evaporation) and then depositing some amount of pheromone on selected solution components. The role of pheromone evaporation is to allow forgetting of 'poor' previous decisions and to help bias the search around the best solutions found. The role of the pheromone deposit is to favour the components of the best solutions in the generation of subsequent solutions.

If ACO is applied to MOOPs, non-dominated solutions are kept in an archive (step 2d) and all non-dominated solutions found during the search are returned. For this purpose, new non-dominated solutions are added to the archive and the solutions that become dominated are removed.

3.4. *Algorithmic components of MOACO algorithm for TSCP*

When applying ACO algorithms to specific problems and to MOOPs in particular, a number of implementation choices have to be made (Dorigo and Di Caro 1999, Dorigo and Stützle 2010, García-Martinez *et al.* 2007, Angus and Woodward 2009, López-Ibáñez and Stützle 2009, 2010). In the following, the details of the underlying ACO algorithm used, *i.e.* MAX-MIN Ant System, are presented, and then the algorithmic components considered to extend this underlying ACO algorithm to tackle the multi-objective version of the TSCP are discussed.

MAX-MIN ant system for TSCP

As for the ACO algorithm, on which the MOACO algorithm for the TSCP is based, the MAX-MIN ant system (MMAS) is used (Stützle and Hoos 1998, 2000). MMAS constructs solutions probabilistically by a standard construction mechanism that is common to most ACO algorithms. An ant starts with an empty solution and at each construction step it chooses probabilistically one solution component. In the classical action choice rule of ACO algorithms, which is also used in the present approach, an ant m chooses solution component (i,j) with a probability given by:

$$p_{ij}^{m} = \frac{\tau_{ij}}{\sum_{l \in \aleph_i^l} [\tau_{il}]} \; \textit{if } j \in \aleph_i^m \tag{1}$$

where $\aleph_i^m$ is the feasible neighbourhood of ant m (*i.e.* the screw elements that are still available in the TSCP case).

Once all ants have constructed a solution, the pheromone trails are updated. The update of the pheromone trails consists of the pheromone trail evaporation, which decreases the amount of pheromone by a fixed factor ρ, and the pheromone deposit. From a high level perspective, this pheromone update process can be described by:

$$\tau_{ij} \rightarrow (1-\rho)\tau_{ij} + \Delta_{ij} \tag{2}$$

In the pheromone deposit, solution components occurring in one or several solutions generated by the ants increase their associated pheromone trail values. This in turn increases the probability that these solution components will be chosen subsequently by the ants in the following iterations. For the pheromone deposit, a first choice has to be made about which solutions are allowed to deposit pheromone on their solution components. One possibility is to consider only the best solution that is generated in the current iteration (*iteration-best strategy*). Another possibility is to consider all the solutions generated since the start of the algorithm and to choose only the best of these (*best-so-far strategy*). Clearly, the latter option results in a more directed search. The amount of pheromone deposited can then be defined, for example, in dependence of the quality of the solution generated.

A particularity of MMAS is that it restricts the level of the allowed pheromone trail values to a range $\tau_{\min} < \tau_{ij} < \tau_{\max}$. Using the pheromone trail limits reduces the possibility of search stagnation around the best solutions and favours the exploration of the search space (Stützle and Hoos 1998, 2000). Following the general rationale of setting these values as explained in Stützle and Hoos (2000), the values of $\tau_{\min}$ and $\tau_{\max}$ are set as follows:

$$\tau_{\max} = \frac{f_i}{\rho} \tag{3}$$

$$\tau_{\min} = \frac{\tau_{\max}}{2loc} \tag{4}$$

$$\tau_{ini} = \frac{\tau_{\max} + \tau_{\min}}{2} \tag{5}$$

where f_i is the best value of the objective function i, loc is the number of screw elements whose location must be defined, and τ_{ini} is the initial pheromone value.

Finally, for applying MMAS to the TSCP, it is also necessary to define the exact meaning of solution component (i, j) and, hence, how the pheromone trail information τ_{ij} is to be interpreted. In the case of the TSCP, a solution component may refer to an assignment of a screw element j to a specific position i on the screw axis (position assignment). In this case, τ_{ij} represents the desirability that screw element j is assigned to position i in the screw axis. Alternatively, a solution component (i, j) may refer to the successor relationship between screw elements on the screw axis (*relation assignment*); in this case, each entry τ_{ij} of the pheromone matrix represents the desirability that element j follows immediately after element i on the screw axis.

Components of the MOACO algorithm

When applying ACO algorithms to MOOPs, a number of generic algorithmic components specific to the multi-objective aspects of the problem need to be defined. In the following, the algorithmic components considered here are discussed.

Number of pheromone matrices and solution construction. When applying ACO to MOOPs, one may use either a single pheromone trail matrix to consider solutions for all objectives or multiple pheromone matrices (García-Martinez *et al.* 2007, López-Ibáñez and Stützle 2009). (In the latter case, typically one pheromone trail matrix per objective is used.) The first option implies that the construction of the solutions follows the usual steps of the ACO algorithm, using the same probabilistic construction rules as in ACO algorithms for single-objective problems, which were presented in the previous subsection.

When using several pheromone matrices, the information contained in the pheromone matrices needs to be aggregated for the probabilistic solution construction. Using a common generic form of this aggregation, an ant m then chooses a solution component (i, j) with a probability given by (Iredi *et al.* 2001, Doerner *et al.* 2004, López-Ibáñez and Stützle 2009, 2010):

$$p_{ij}^{m} = \frac{\tau_{ij}^{\lambda_1} \otimes \tau_{ij}^{\lambda_2}}{\sum_{l \in \aleph_i^l} [\tau_{il}^{\lambda_1} \otimes \tau_{il}^{\lambda_2}]} \; if \; j \in \aleph_i^m \tag{6}$$

where $\otimes$ is an aggregation operator, and λ_1 and λ_2 are the weights attributed to each objective, if two objectives are considered. The weights are normalized such that $\sum_i \lambda_i = 1$. [Equation (6) can easily be extended to more objectives.] Thus, the exploration of different regions of the Pareto front can be assured by the assignment of different weights to the ants. The proposed algorithm is always assumed to have a maximally dispersed set of weight vectors.

Concerning the aggregation operator, two different choices have been taken in previous multi-objective ACO (MOACO) algorithms. One possibility is to use a weighted product aggregation (*product strategy*), *i.e.* in Equation (6) the operator $\otimes$ is actually $\times$. A second possibility is to use a weighted sum aggregation (*linear strategy*), *i.e.* in Equation (6) the operator $\otimes$ is replaced by $+$. Both choices were examined in the experimental analysis performed here.

Pheromone update. The pheromone update generally follows the steps explained above for MMAS. The main issue to be considered in the multi-objective case is the choice of which solutions are used for the pheromone deposit.

Independently of whether one uses the iteration-best or the best-so-far strategy, the choice of the solutions for pheromone deposit depends on the number of pheromone matrices considered. If a *single* pheromone trail matrix for all objectives exists, the pheromone update is performed by using a maximum number of k non-dominated solution (or ants). Each of the k ants deposits an amount $1/k$ of pheromone. The set of k non-dominated ants is selected through the use of a niching mechanism, in order that the solutions become evenly distributed along the Pareto front approximation (Deb and Goldeberg 1989). (To reflect the choice of the amount of pheromone to be deposited in this case, in the definition of τ_{max} for the underlying MMAS algorithm, τ_{max} is set equal to 2.0.) If *multiple* pheromone matrices are used, only a single pheromone matrix for each objective is used. In this case, the pheromone is deposited on the best solution (*i.e.* on the iteration-best or the best-so-far solution, depending on which strategy is chosen) for each objective. The amount of pheromone deposited is then chosen to be proportional to the quality of the solutions.

Number of colonies. Multiple colonies can be used in MOACO algorithms to better distribute the search effort along the Pareto front (Iredi *et al.* 2001). Each colony is then specialized to search in a specific area of the Pareto front. For this purpose, the m ants are divided into p colonies. Each colony has its own pheromone information and the ants of each colony construct the solutions based only on their colony's pheromone information. The colonies then cooperate through the pheromone update mechanism. For example, the non-dominated solutions may be determined by considering the solutions generated by all colonies, imposing in this way a more selective determination of the non-dominated set of solutions.

In the pheromone update, cooperation can be achieved by the pheromone *update by region* scheme (Iredi *et al.* 2001). In the bi-objective case considered here, the non-dominated solutions are first sorted along the front and then split into p subsets (p is the number of subsets, which is equivalent to the number of colonies). The best solutions of subset j are then used to update the pheromone information of colony *j*. In contrast, in the *update by origin* approach (Iredi *et al.* 2001), the ants of a specific colony are only able to update the pheromone trails of their own colony. Here, the update by region approach is adopted, since it is deemed to induce a more directed search.

The use of several colonies can be combined with MOACO approaches that use a single pheromone trail matrix or those that use several pheromone trail matrices. In this article, the focus is given to the second possibility. (*A posteriori*, this choice is also justified by the fact that in the experimental analysis presented in Section 5 it turned out that the use of one pheromone trail matrix per objective resulted in better performance when compared to using only a single pheromone matrix.) When combining pheromone trail matrices by weights (λ several possibilities for distributing the weights among colonies have been suggested (Iredi *et al.* 2001). Here, 50% of the weights attributed to colony k are overlapped by the weights of colony $k + 1$ and $k - 1$.

4. Case studies

The experiments are based on the characteristics of a Leistritz co-rotating twin-screw extruder, which is available at the University of Minho (Gaspar-Cunha *et al.* 2011). Table 3 shows the length and pitch of the screw elements used for each of the four instances considered. A polypropylene homopolymer (ISPLEN PP 030G1E from Repsol) is being processed using a barrel and die temperature set to 220 °C, the screws rotating at 150 rpm and a feed rate of 8 kg/h. The goal is to define the best location of the 16 screw elements identified in Table 3 such that performance is optimized. The screw elements available are conveying screw elements with different lengths and pitches (20, 30, 45 and 60 mm), a single left-handed element and kneading blocks with different staggering angles (−30°, 45° and −60°). The number of restrictive screw elements increases progressively from one (instance TSCP1) to four (instance TSCP4). In order to guarantee enough initial conveying capability, the two first conveying screw elements in the screw axis are fixed; this means that the location of the 14 other elements needs to be defined by the optimization algorithms. For each of these four instances, three case studies defined by the combination of the pairs of objectives presented in Table 4 are examined. Thus, a total of 12 case studies (*i.e.* optimization sets) is studied.

As stated before, the main goal of this work is to develop a high-performing MOACO algorithm for the multi-objective TSCP. For this purpose, first the several design alternatives of MOACO algorithms are tested, which have been discussed in the previous section and are summarized in Table 5. Then, the performance of the final MOACO algorithm to the previously developed RPSGA and a TPLS algorithm are compared. Since the computation time of the modelling routine for evaluating a solution is relatively high (between 2 and 3 min, depending on screw complexity), each algorithm run was limited to 3000 solution evaluations.

Table 3. Screw elements used in each of the four instances TSCP1 to TSCP4.

Instance		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
TSCP1	Length	97.5	120	45	60	30	30	30	60	**30**	120	30	120	37.5	60	60	30
	Pitch	45	30	45	30	20	60	30	20	**KB–60**	30	30	60	20	45	30	20
TSCP2	Length	97.5	120	45	60	**30**	30	30	60	**30**	120	30	120	37.5	60	60	30
	Pitch	45	30	45	30	**–20**	60	30	20	**KB–60**	30	30	60	20	45	30	20
TSCP3	Length	97.5	120	**45**	60	**30**	30	30	60	**30**	120	30	120	37.5	60	60	30
	Pitch	45	30	**KB–45**	30	**–20**	60	30	20	**KB–60**	30	30	60	20	45	30	20
TSCP4	Length	97.5	120	**45**	60	**30**	30	30	60	**30**	120	30	120	**37.5**	60	60	30
	Pitch	45	30	**KB–45**	30	**–20**	60	30	20	**KB–60**	30	30	60	**KB–30**	45	30	20

Table 4. Case studies for each of the four instances of Table 1. Given are the optimization criteria, the aim of optimization and the feasible ranges of the objectives.

Case study	Criteria	Aim	X_{min}	X_{max}
1	Average strain	Maximization	1000	15000
	SME	Minimization	0.1	2
2	Average strain	Maximization	1000	15000
	Viscous dissipation	Minimization	0.9	1.5
3	SME	Minimization	0.1	2
	Viscous dissipation	Minimization	0.9	1.5

Note: SME = specific mechanical energy.

Table 5. Components of the MOACO algorithm that have been studied.

MOACO component	Values tested
Pheromone evaporation	0.1, 0.2, 0.3, 0.4 and 0.5
Pheromone information	Position *vs* relation
Pheromone update strategy	Best-so-far *vs* iteration-best
Number of pheromone matrices	One *vs* several
Number of colonies	One *vs* three
Weight aggregation	Linear *vs* product

Given the stochastic nature of the optimizers, it is necessary to adopt statistical methods to compare the non-dominated solutions produced—the use of a single value would have severe limitations (Zitzler *et al.* 2003, Knowles *et al.* 2006). For this reason, the performance of the algorithms was assessed using the empirical attainment function (EAF) methodology (Grunert da Fonseca *et al.* 2001). In simple terms, the EAF estimates the probability for a particular point in the objective vector to be attained by a single run of the algorithm (Fonseca and Fleming 1996). The EAFs are measured by running the multi-objective algorithms a number of times; here, 10 independent runs are carried out with each algorithm. The performance of two algorithms can then be compared through plots of the differences of their EAFs (Lópes-Ibáñez *et al.* 2006, 2010), which are indicated by points in the objective space. The points are only plotted in positions where the differences between the EAFs of the two algorithms change between different values.

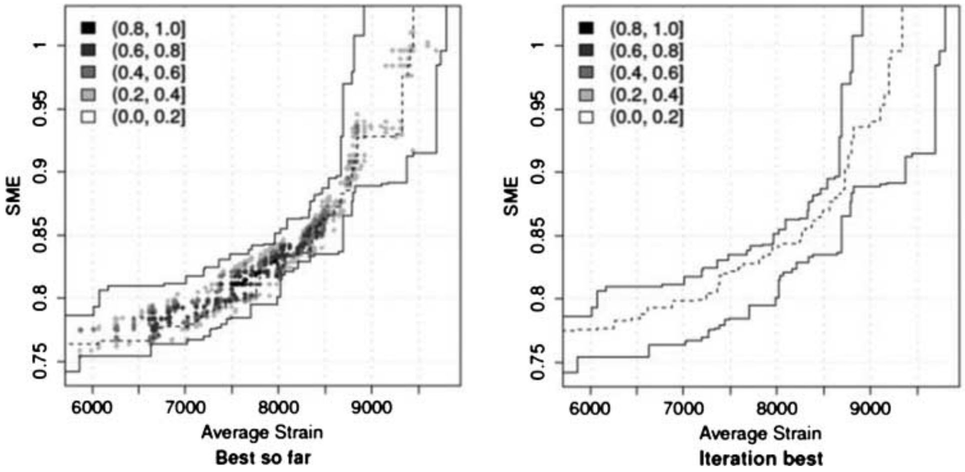

Figure 4. Influence of the pheromone update strategy (best-so-far *vs* iteration-best). SME = specific mechanical energy.

The values of the differences are encoded using a grey scale; the darker the points, the larger the observed differences. An example of such a plot is given in Figure 4. The two continuous border lines in the plot connect the best objective vectors that have been found in any of the runs of the two algorithms (grand best) and the points in the objective space that have been dominated by all runs of the two algorithms (grand worst); the discontinuous line in the middle gives the median of the EAF that was obtained by an algorithm (*e.g.* in Figure 4, the left plot is given the median obtained by the best-so-far pheromone update strategy), *i.e.* the boundary of the objective space that was obtained in 50% of the algorithm runs. Figure 4 clearly indicates a better performance of the MOACO algorithm with the best-so-far pheromone update strategy compared to the iteration best pheromone update strategy. In fact, all the differences between the EAFs are in favour of the best-so-far pheromone update strategy, some being rather large (between 0.8 and 1.0 as indicated by the black points).

5. Results and discussion

First, the influence of the MOACO design choices and parameter settings on performance, using the three case studies associated to instance TSCP4, is analysed. Then, the best MOACO results to RPSGA and TPLS on all 12 case studies (using all instances TSCP1 to TSCP4) are compared. In the experiments presented here, the colonies used a cross-total of 50 ants (except in the case where the influence of the number of colonies is studied) and 60 iterations were performed, resulting in 3000 solution evaluations with the modelling routine.

5.1. *Influence of MOACO components*

In this section, a summary of the main results obtained when comparing different design choices and parameter settings of the MOACO algorithm is presented. Since the presentation of all EAF differences plots would be too extensive, only a few illustrative results using instance TSCP4 and case studies 1 and 2 (see also Table 3) are reported here; the complete set of results is available at http://www.dep.uminho.pt/agc/agc/Supplementary_Information_Page.html.

Pheromone information

The comparison between the relation and the position-based pheromone information (Figure 5 of the supplementary file) using the plots of the EAF differences demonstrate that almost all differences are in favour of the relation-based definition of the pheromone information. These are expected results, since in the polymer extrusion process studied here there is a strong interaction between contiguous elements, mainly when restrictive elements are considered.

The performance of the position-based pheromone information could also not be further improved by the application of additional techniques. For example, the application of the pheromone summation rule is considered (Merkle and Middendorf 2000), but no performance improvement was obtained (plots shown in the supplementary material, Figure 10). As a further refinement, the choice of an assignment order is considered, which gives preference to first assign screw elements (*e.g.* kneading discs and left handed elements), which are supposed to have a more important impact on the global process characteristics than others. However, in the tests on the assignment order of the screw elements (sequential *vs* importance-based), a consistent improvement using this ‘importance-based’ assignment order could not be obtained. Hence, neither the summation rule nor the importance-based assignment order was included in the

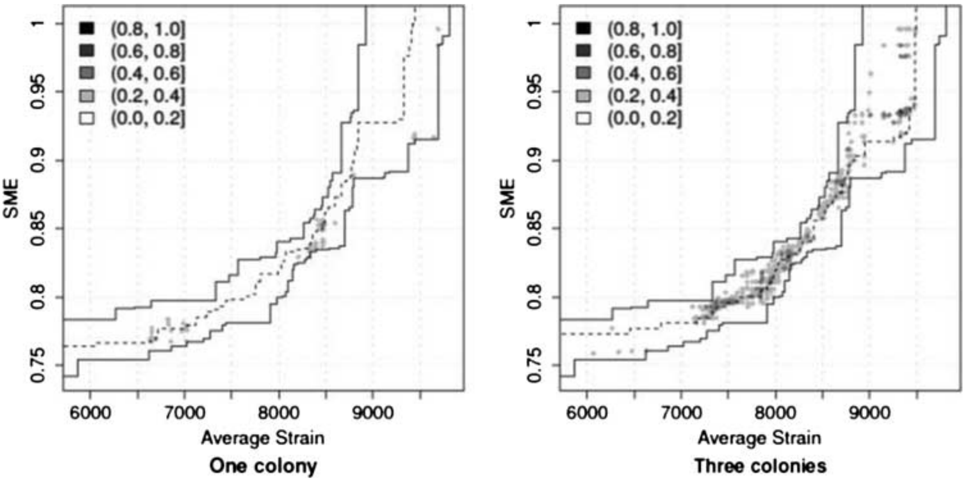

Figure 5. Influence of the number of colonies (one colony *vs* three colonies). SME = specific mechanical energy.

final MOACO algorithm. For the following results, the relation-based definition of the pheromone information is always used.

Pheromone update strategy

Figure 4 presents the results obtained for the two different pheromone update strategies tested. The best-so-far update strategy (best among the solutions existing in the archive) produces much better results than the iteration-best update strategy and, hence, it will be adopted in other experiments. The influence of the pheromone trail evaporation factor ρ (shown in the supplementary pages) is also examined. The best results were obtained for values of $\rho = 0.2$ and $\rho = 0.3$.

Number of pheromone matrices

As anticipated, the utilization of several pheromone matrices, using one matrix for each objective together with weighted aggregation of the pheromone trail values, produced significantly better results than the adoption of a single pheromone trail matrix (see Figure 9 in the supplementary information file).

Number of colonies

Two different conditions were analysed: the application of a single colony with 50 ants during 60 iterations and the application of three colonies with 20 ants each during 50 iterations. In both cases, each colony uses two pheromone matrices. (Note that for each of the three colonies 20 ants and 50 iterations are used to limit the computational effort to 3000 solution evaluations by the modelling routine.) Better results were attained in the second case, as depicted in Figure 5.

Weight aggregation

Finally, as seen in Figure 6, it can be concluded that the product strategy for the pheromone aggregation produces substantially better results than the weighted sum aggregation by the linear method.

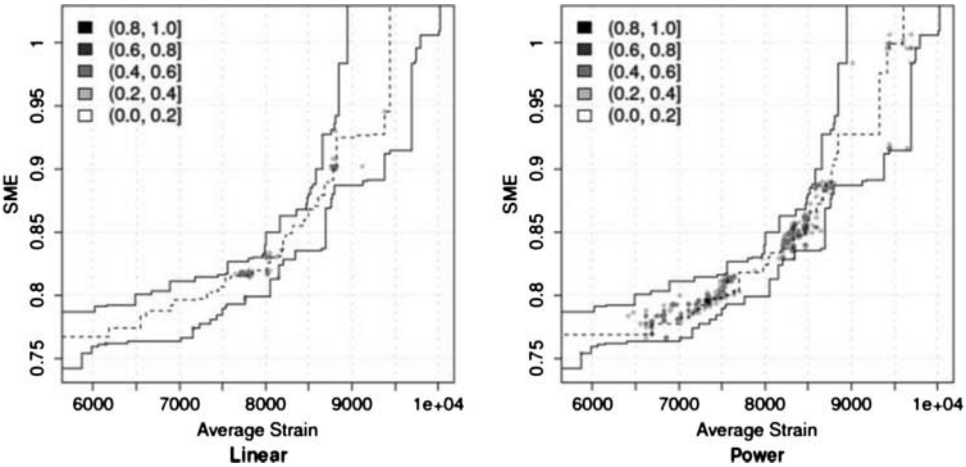

Figure 6. Influence of the weight aggregation method (linear *vs* product). SME = specific mechanical energy.

Table 6. Best parameters values suggested from experimental analysis for the MOACO algorithm.

MOACO component	Best value
Pheromone evaporation rate	0.2
Pheromone information	Relation
Pheromone update strategy	Best-so-far
Number of pheromone matrices	Several
Number of colonies	Three
Weight aggregation	Product

A summary of the above results is presented in Table 6. The second column represents the best values obtained when the parameters are studied individually. These best values are used for the final MOACO algorithm, the performance of which is compared with the RPSGA in the next section.

5.2. *Comparison with RPSGA and TPLS*

Here, the performance of the MOACO algorithm (using the best combination of parameters, as presented in Table 6) is compared with RPSGA and TPLS. Each algorithm is run for a maximum of 3000 solution evaluations on each of the 12 case studies of the bi-objective TSCP.

The differences among the empirical EAFs can be observed in Figure 7 for the case of the TSCP4 instance (the corresponding figures for instances TSCP1 to TSCP3 can be consulted in the Supplementary Information Page repository: http://www.dep.uminho.pt/agc/agc/Supplementary_Information_Page.html; Figures 23, 24 and 25, respectively), where each figure relates to the results comparing the three case studies for the same number of restrictive elements to be sequenced (see also Table 3). The differences between the algorithms are higher for the case studies on TSCP3 and TSCP4 with three and four restrictive elements, respectively. In fact, in this case the advantages are almost exclusively in favour of the MOACO algorithm. On the case studies related to TSCP1 and TSCP2, still most of the differences are in favour of the MOACO algorithm, although they are typically limited to smaller areas of the objective space. This outcome

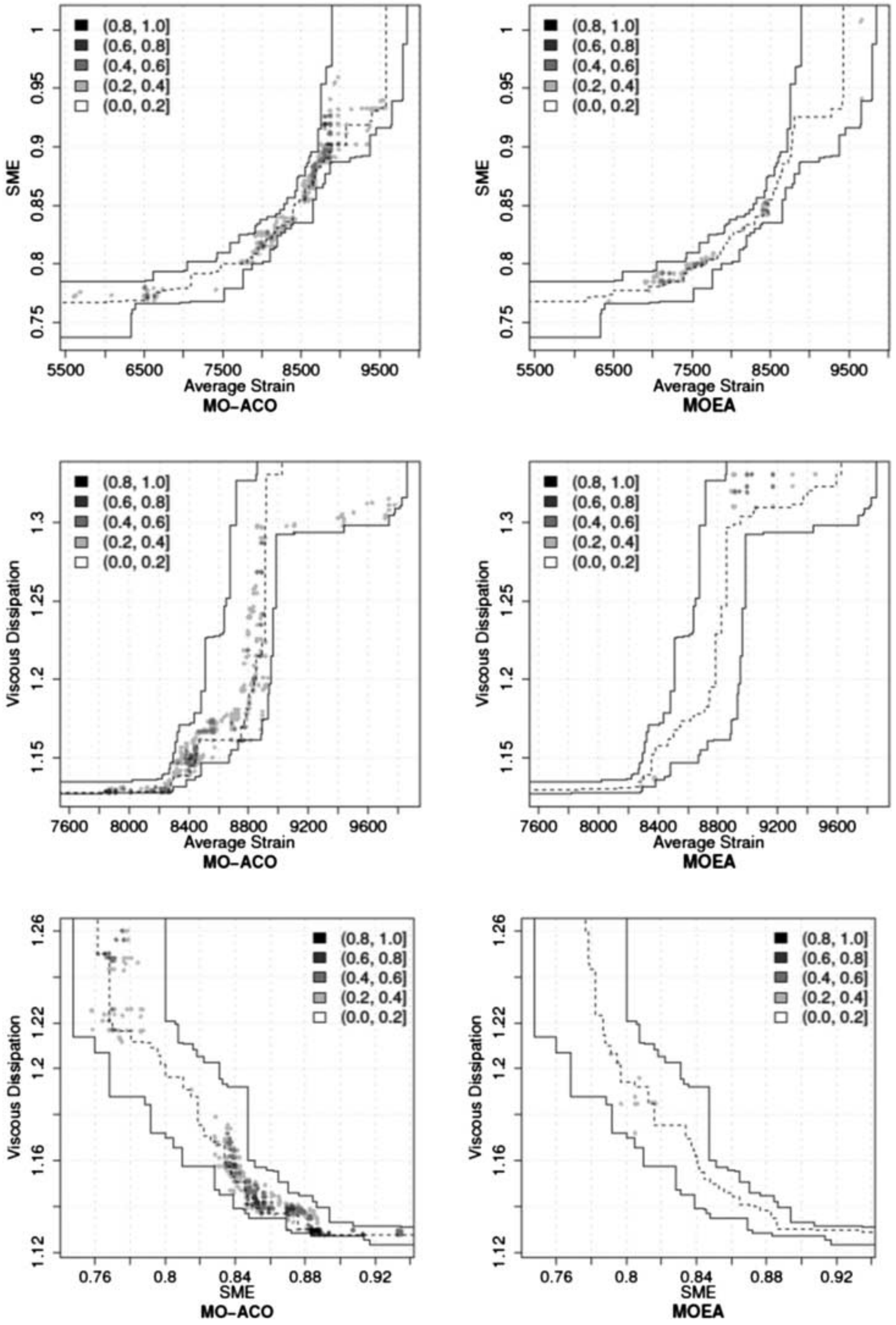

Figure 7. Comparison between MOACO and RPSGA for TSCP4 instance. SME = specific mechanical energy.

seems to indicate that the advantage of MOACO over RPSGA increases with a growing number of restrictive screw elements.

As a further analysis, the hypervolume indicator was also computed (Zitzler *et al.* 2003). For minimization problems, the hypervolume in two dimensions measures the surface that is

Table 7. Comparison of normalized hypervolume values between RPSGA and MOACO.

Instance	Objectives	MOACO	RPSGA	*p*-Value
TSCP1	Average strain, SME	*0.9502*	0.9421	0.1403
	Average strain, viscous dissipation	*0.9039*	0.8777	0.9705
	SME, viscous dissipation	***0.6563***	0.6413	1.083e-05
TSCP2	Average strain, SME	***0.8432***	0.8147	0.0311
	Average strain, viscous dissipation	*0.7570*	0.7431	0.2526
	SME, viscous dissipation	*0.7522*	0.7418	0.6772
TSCP3	Average strain, SME	***0.8554***	0.8070	0.0041
	Average strain, viscous dissipation	***0.8156***	0.7368	0.0039
	SME, Vis. Dissipation	*0.6975*	0.6805	0.1655
TSCP4	Average strain, SME	*0.7618*	0.7439	0.3073
	Average strain, viscous dissipation	*0.5361*	0.5042	0.2475
	SME, viscous dissipation	*0.6563*	0.6154	0.0630

Note: An entry marked in italic face indicates a higher hypervolume; an entry marked in bold face indicates a statistically significant difference according to the Wilcoxon signed rank test in favour of one algorithm. In the last column the observed *p*-values for the Wilcoxon test are shown.
SME = specific mechanical energy.

dominated by the non-dominated solutions of a Pareto-front approximation and bounded by a point that is larger in every objective than any of the solutions in the Pareto-front approximation. The case of maximization problems can be described analogously. Given the very large differences in the ranges between the three objectives, all results are first normalized into the interval [1,2]. This is done by first removing dominated points, then converting any maximization objective into a minimization one, and finally considering for each objective and each instance the smallest and largest objective values found by any of the two algorithms among the non-dominated solutions. The smallest value is then mapped to 1.0 and the largest to 2.0. The hypervolume is then computed taking the worst point, (2.0,2.0), as a reference. Using the resulting hypervolume values, a Wilcoxon signed rank test was conducted to examine the statistical significance of the observed differences. The results are summarized in Table 7, where the average hypervolume values for each of RPSGA and MOACO measured across the 10 independent runs of each algorithm and the *p*-value of the statistical test are presented. The larger of the two hypervolume values is indicated in italic; if the differences are significant at the usual 0.05 significance level, this is noted in bold. As can be seen from these results, in all case studies, the MOACO algorithm obtains a higher average hypervolume value than RPSGA, indicating its superiority over RPSGA. In fact, a binomial test, which has as its null hypothesis that RPSGA and MOACO have the same probability of reaching a smaller average hypervolume than the other, is rejected in favour of MOACO ($p = 0.00024$). Hence, it is possible to conclude that, for the case studies considered, MOACO reaches better performance than RPSGA. If each individual comparison on each case study is given attention in turn, it is possible to observe that in four of the 12 case studies, the difference in hypervolume is statistically significant in favour of MOACO (and none is in favour of RPSGA). This is noteworthy, since by having only 10 independent runs of each algorithm, the power of the test is not very high.

Finally, the performance of the MOACO algorithm was compared with the earlier results obtained with the TPLS algorithm (Teixeira *et al.* 2011). (To limit the length of this article, the detailed results with the plots of EAF differences can be found in the online available supplementary pages in Figures 27–30). The same conclusions can be drawn, *i.e.* MOACO is superior to its competitor in most case studies. (In fact, only for one case study—instance TSCP4, case study 1, involving average strain and SME objectives—does TPLS have an advantage over the MOACO algorithm.) Hence, the proposed MOACO algorithm is the current method of choice to tackle the TSCP.

6. Conclusion

An effective MOACO algorithm for the bi-objective TSCP is presented. First, several different design choices and parameter settings of the MOACO algorithm were examined. Based on the knowledge gained with that study, a MOACO algorithm for the bi-objective TSCP is proposed and its performance is evaluated on a total of 12 case studies. The analysis of the experimental results via the use of state-of-the-art methods for evaluating the performance of multi-objective optimizers demonstrated that the proposed MOACO algorithm is the method of choice for tackling the TSCP. Effective algorithmic tools to define appropriate screw configurations are indeed required by the industry.

This research may be extended in a number of possible directions. The first is to consider combinations of MOACO algorithms with TPLS, or other local search methods, to further improve the results. Given the high computation times required by the modelling routine to evaluate the screw configurations, another promising direction would be to reduce the number of calls to the modelling routine by using statistical modelling techniques for predicting the objective function values. Finally, extending the work to embrace the situation where the user must choose the appropriate screw elements from a larger set of available elements would be very interesting from a practical standpoint. In this case, two interrelated problems arise. The first is selecting the screw elements from a larger set of available ones, while the second is defining the sequence of elements on the screw shaft.

Acknowledgements

This work has been supported by the Portuguese Fundação para a Ciência e Tecnologia under PhD grant SFRH/BD/21921/2005. Thomas Stützle acknowledges support of the Belgian F.R.S-FNRS of which he is a research associate, the E-SWARM project, funded by an ERC Advanced Grant, and by the Meta-X project, funded by the Scientific Research Directorate of the French Community of Belgium.

References

Angus, D. and Woodward, C., 2009. Multiple objective ant colony optimization. *Swarm Intelligence*, 3 (1), 69–85.

Coello Coello, C.A., Lamont, G., and Van Veldhuizen, D.A., 2007. *Evolutionary algorithms for solving multi-objective problems*. New York: Springer.

Deb K., 2001. *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.

Deb, K. and Goldberg, D.E., 1989. An Investigation of niche and species formation in genetic function optimization. *In*: *Proceedings of the third international conference on genetic algorithms*. San Francisco, CA: Morgan Kauffman, 41–49.

Deb, K., *et al.*, 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGAII. *In*: M. Schoenauer *et al.*, eds., *Parallel problem solving from nature (PPSN-VI). Lecture notes in computer science*, Vol. 1917. Heidelberg: Springer, 849–858.

Doerner, K., *et al.*, 2004. Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 131 (3), 79–99.

Domingues, N., Gaspar-Cunha, A., and Covas, J.A., 2010. Estimation of the morphology development of immiscible liquid–liquid systems during single screw extrusion. *Polymer Engineering and Science*, 50 (11), 2194–2204.

Dorigo, M. and Di Caro, G., 1999. The ant colony optimization metaheuristic. *In*: D. Corne, M. Dorigo and F. Glover, eds. *New ideas in optimization*. London: McGraw-Hill, 1–32.

Dorigo, M. and Stützle, T., 2010. Ant colony optimization: overview and recent advances. *In*: M. Gendreau and Y. Potvin, eds. *Handbook of metaheuristics*. 2nd ed. *International series in operations research & management science*, Vol. 146. New York: Springer, 227–263.

Fonseca, C.M. and Fleming, P., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. *In*: H.M. Voigt *et al.*, eds. *Proceedings of fourth international conference on parallel problem solving from nature (PPSN-IV). Lecture notes in computer science*, Vol. 1141. Heidelberg: Springer, 584–593.

García-Martínez, C., Córdon, O., and Herrera, F., 2007. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180 (1), 118–148.

Gaspar-Cunha, A., 2000. *Modelling and optimization of single screw extrusion*. Thesis (PhD). University of Minho, Guimarães, Portugal.

Gaspar-Cunha, A. and Covas, J.A., 2004. RPSGAe—a multiobjective genetic algorithm with elitism: application to polymer extrusion. *In*: X. Gandibleux *et al.*, eds. *Metaheuristics for multiobjective optimization. Lecture notes in economics and mathematical systems*, Vol. 535. Berlin: Springer, 221–249.

Gaspar-Cunha, Oliveira, P. and Covas, J.A., 1997. Use of genetic algorithms in multicriteria optimization to solve industrial problems. *In*: T. Bäck, ed. *Seventh international conference on genetic algorithms*. San Francisco, CA: Morgan Kaufmann, 682–688.

Gaspar-Cunha, A., *et al.*, 2002. Optimization of processing conditions for polymer twin-screw extrusion. *International Polymer Processing*, 17 (3), 201–213.

Gaspar-Cunha, A., Covas, J.A., and Vergnes, B., 2005. Defining the configuration of co-rotating twin screw extruders with multiobjective evolutionary algorithms. *Polymer Engineering and Science*, 45 (8), 1159–1173.

Gaspar-Cunha, A., *et al.*, 2011. Reactive extrusion—optimization of representative processes. *In*: A. Gaspar-Cunha and J.A. Covas, eds. *Optimization in polymer processing*. New York: Nova.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimisation and machine learning*. Reading, MA: Addison-Wesley.

Grunert da Fonseca, V., Fonseca, C.M., and Hall, A., 2001. Inferential performance assessment of stochastic optimizers and the attainment function. *In*: E. Zitzler *et al.*, eds. *Evolutionary multicriterion optimization (EMO 2001). Lecture notes in computer science*, Vol. 1993. Heidelberg: Springer, 213–225.

Iredi, S., Merkle, D., and Middendorf, M., 2001. Bicriterion optimization with multi-colony ant algorithms. *In*: E. Zitzler *et al.*, eds. *Evolutionary multicriterion optimization (EMO 2001). Lecture notes in computer science*, Vol. 1993. Heidelberg: Springer, 359–372.

Knowles, J., Thiele, L., and Zitzler, E., 2006. *A tutorial on the performance assessment of stochastive multiobjective optimizers.* Computer Engineering and Networks Laboratory, ETH Zurich. Technical Report TIK-Report No. 214.

Kohlgrüber, K., 2007. *Co-rotating twin-screw extruders: fundamentals, technology, and applications*. Munich: Hanser.

López-Ibáñez, M. and Stützle, T., 2009. An analysis of algorithmic components for multiobjective ant colony optimization: a case study on the biobjective TSP. *In*: P. Collet *et al.*, eds. *Artificial evolution—9th international conference, evolution artificielle (EA 2009). Lecture notes in computer science*, Vol. 5975. Heidelberg: Springer, 134–145.

López-Ibáñez, M. and Stützle, T., 2010. The impact of design choices of multiobjective ant colony optimization algorithms on performance: an experimental study on the biobjective TSP. *In*: *Proceedings of the genetic and evolutionary computation conference (GECCO 2010).* New York: ACM Press, 71–78.

López-Ibáñez, M., Paquete, L., and Stützle, T., 2006. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. *Journal of Mathematical Modelling and Algorithms*, 5 (1), 111–137.

López-Ibáñez, M., Paquete, L., and Stützle, T., 2010. Exploratory analysis of stochastic local search algorithms in biobjective optimization. *In*: T. Bartz-Beielstein, *et al.*, eds. *Experimental methods for the analysis of optimization algorithms.* Berlin: Springer, 209–233.

Mariano, C. and Morales, E., 1999. *A multiple objective ant-Q algorithm for the design of water distribution irrigation networks*. Instituto Mexicano de Tecnologia Del Agua. Technical Report HC-9904.

Merkle, D. and Middendorf, M., 2000. An ant colony with a new pheromone evaluation rule for tardiness problems. *In*: S. Cagnoni *et al.*, eds. *Proceedings of the Evo workshops 2000. Lecture notes in computer science*, Vol. 1803. Heidelberg: Springer, 287–296.

Potente, H. and Thumen, A., 2006. Method for the optimization of screw elements for tightly intermeshing, co-rotating twin screw extruders. *International Polymer Processing*, 21 (2), 149–154.

Sakai, T., 1991. Report on the state of the art: reactive processing using twin-screw extruders. *Advances in Polymer Technology*, 11 (2), 99–108.

Stützle, T. and Hoos, H., 1998. Improvements on the ant system: introducing the Max-Min ant system. *In*: G.D. Smith, N.C. Steele and R.F. Albrecht, eds. *Proceedings of artificial neural nets and genetic algorithms*. Vienna: Springer, 245–249.

Stützle, T. and Hoos, H., 2000. MAX-MIN ant system. *Future Generation Computer Systems*, 16 (8), 889–914.

Tao, G. and Michalewicz, Z., 1998. Inver-over operator for the TSP. *In*: T. Baeck, *et al.*, eds. *Parallel problem solving from nature (PPSN V).* Heidelberg: Springer, 803–812.

Teixeira, C., *et al.*, 2007. Modelling flow and heat transfer in co-rotating twin-screw extruders. *In*: E. Cueto and F. Chinesta, eds. *10th Esaform conference on material forming*, *AIP conference proceedings*, Vol. 907, Zaragoza, Spain. American Institute of Physics, 980–985.

Teixeira, C., *et al.*, 2010. Application of evolutionary algorithms to the definition of the optimal twin-screw extruder configuration for starch cationization. *Polymer Engineering and Science*, 51 (2), 330–340.

Teixeira, C., *et al.*, 2011. Engineering an efficient two-phase local search for the co-rotating twin-screw configuration problem. *International Transactions in Operational Research*, 18 (2), 271–291.

Vergnes, B., Della Valle, G., and Delamare, L., 1998. A global computer software for polymer flows in corotating twin screw extruders. *Polymer Engineering and Science*, 38 (11), 1781–792.

White, J., *et al.*, 2001. *Co-rotating twin screw extruder modeling program* [online]. Available from: http://www.temarex.com/extrus.htm [Accessed 28 January 2011].

White, J.L. and Kim, E.K., 2010. *Twin screw extrusion, technology and principles*. 2nd ed. Munich: Hanser.

Zitzler, E., *et al.*, 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7 (2), 117–132.