

# Sequential Ordering Algorithm for Mass Integration: The Case of Direct Recycling

Filippo Marchione<sup>1</sup>, Stavros Papadokostantakis<sup>2\*</sup>, Konrad Hungerbuehler<sup>1</sup>

<sup>1</sup>Department of Chemistry and Applied Biosciences, Swiss Federal Institute of Technology, Zurich, Switzerland

<sup>2</sup>Department of Energy and Environment, Chalmers University of Technology, Gothenburg, Sweden

Email: \*stavros.papadokostantakis@chalmers.se

Received 4 February 2016; accepted 18 April 2016; published 21 April 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

In the last three decades much effort has been devoted in process integration as a way to improve economic and environmental performance of chemical processes. Although the established frameworks have undergone constant refinement toward formulating and solving complicated process integration problems, less attention has been drawn to the problem of sequential applications of mass integration. This work addresses this problem by proposing an algorithm for optimal ordering of the process sinks in direct recycling problems, which is compatible with the typical mass integration formulation. The order consists in selecting the optimal sink at a specific integration step given the selection of the previous steps and the remaining process sources. Such order is identified through a succession of preemptive goal programming problems, namely of optimization problems characterized by more objectives at different priority levels. Indeed, the target for each sink is obtained by maximizing the total flow recycled from the available process sources to this sink and then minimizing the use of pure sources, starting from the purest one; the hierarchy is respected through a succession of linear optimization problems with a single objective function. While the conditional optimality of the algorithm holds always, a thorough statistical analysis including structured to random scenarios of process sources and process sinks shows how frequently the sequential ordering algorithm is outperformed with respect to the total recycled amount by a different selection of process sinks with the same cardinality. Two more case studies proving the usefulness of ordering the process sinks are illustrated. Extensions of the algorithm are also identified to cover more aspects of the process integration framework.

## Keywords

Process Integration, Preemptive Goal Programming, Conditional Optimality, Recycling Activities Prioritization, Linear Programming

---

\*Corresponding author.

## 1. Introduction

The reduction of the cost and environmental impact of industrial processes is among the most relevant challenges for chemical engineers to design sustainable processes [1] [2]. From this perspective, the focus is typically on reducing the use of raw materials, energy utilities and fresh water, which also typically affects the produced waste loads and, therefore, the cost and the environmental impact associated with the end-of-pipe technologies. This has also motivated the field of process systems engineering to formulate the respective problems and develop efficient techniques to handle their increased complexity [3]. In this regard, process optimization and mathematical programming have undergone a significant development over the last three decades [4], particularly the field of process integration focusing on optimal interaction between process units, process streams and fresh resources [5].

Among the various methods and objectives of process integration, mass integration through direct recycling and mass exchange networks is a methodology that allows recovering part of the waste streams of a process (*i.e.*, process sources) and reusing them in appropriate process units (*i.e.*, process sinks), thus reducing the purchase of fresh resources [6] [7]. Many past works have utilized graphical approaches [8]-[10] and algebraic methods [11]-[13] to calculate the target for maximum recycling and, therefore, minimum fresh resource utilization. Such methods, although originally designed for continuous processes, have also been extended to batch processes [14]-[17].

More recent methodological development in the problem formulation and solution strategies of mass integration has focused on advanced optimization techniques. In this regard, Bagajewicz and Savelski presented a formulation based on mixed integer linear programming (MILP) to solve the water allocation problem in process plants when a single contaminant is present [18]. Gabriel *et al.* proposed an approach to reformulate a general direct recycle/reuse of process waste problem into a linear problem (LP), thus simplifying the calculation of the global solution [19]. The strategies presented by Alva-Argaez *et al.* [20] use superstructure formulation, where the design of water networks is initially expressed in a mixed integer nonlinear programming (MINLP) formulation and then decomposed into MILP for an efficient solution strategy. In the same direction, Faria *et al.* [21] suggested a method for the design of multicomponent wastewater networks, where the discretization of one variable of the nonlinear programming problem (NLP) implies the generation of a MILP, which is finally solved through an iterative procedure. For this purpose, Karupiah *et al.* solved the problem of the optimal synthesis of integrated water networks by first formulating a general superstructure as a nonlinear programming problem, which is then solved through a spatial branch-and-bound algorithm where the nonconvex terms are approximated to obtain a simplified problem and get tight lower bounds on the global optimum [22]. Similarly, Ahmetovic *et al.* presented an optimization strategy for large-scale process water networks expressed as NLP or MINLP, where the bounds on the variables are obtained by solving general equations derived by physical inspection [23].

Recent works have also extended the complexity of the mass integration problem formulation. This includes, for instance, multi objective optimization of waste water integration problems with multiple pollutant substances [24]-[26], simultaneous mass and property integration considering also thermal constraints [27], and the synthesis of optimal water networks in the case of time-variable configurations [28].

In all the approaches mentioned above, the main focus is to find the optimal solution of the respective mass integration problem (e.g., proposing a final optimal design of a direct recycling or mass exchange network) without proposing a sequential order of actions to reach this optimal solution. However, creating an order of integration actions can be of special interest in practical engineering problems where the optimal overall solution cannot be realized at once but only as a sequence of steps. Some reasons can be significant deviations in the actual performance of the integration steps (e.g., because of modeling uncertainties or simplifications), the fact that it may not be *a priori* known if all the necessary integration steps will be executed (e.g., because of time and capital expenses limitations considering also the relative significance of the integration step) or, due to process operability reasons, the transition to the optimal integration design has to be practically sequential (e.g., to avoid production loss through extended shut-down). A further more qualitative reason refers to the enhanced interpretability of the proposed solutions that is often critical for the implementation of new designs in industrial practice.

This work addresses this problem by proposing a new algorithm for optimal ordering of the process sinks in direct recycling mass integration problems. The order consists in selecting the optimal sink at a specific integra-

tion step when the selection of the previous steps and the remaining process sources are given, ensuring also that the overall optimal direct recycling target is reached. The latter is achieved through a succession of preemptive goal programming (PGP) problems, namely of optimization problems characterized by more objectives at different priority levels [29], which in this case are the amount recycled to a process sink and the purity of the sources that have been used for this task. This approach ensures to reach the overall direct recycling target similarly to the sink composition and source prioritization rules proposed by El-Halwagi [6]. This study also identifies conditions, under which, for a particular cardinality of the solution (*i.e.*, number of integration steps), the group of sinks is not only sequentially but also globally optimal (*i.e.*, with respect to all the solutions with the same cardinality). Moreover, a statistical analysis for a wide range of direct recycling scenarios demonstrates how frequently the sequential optimality corresponds also to overall optimality for a given number of integration steps. The usefulness of the ordering algorithm for the process sinks is illustrated in two case studies. The first refers to a sequential “switch off/switch on” operation of process units for realizing a mass integration plan. The second refers to realizing a mass integration plan in process units with relevant nonlinear behavior with respect to the impurities in recycle streams and discusses the potential benefits of the sequential approach proposed herein. Finally, this study also highlights the further needs for methodological research in this direction.

## 2. Methods

### 2.1. Direct Recycling Problem Formulation

From the classic direct recycling perspective, a generic chemical process is characterized by the presence of waste streams comprising a target compound which in presence of several impurities can be partially or fully recovered through direct recycling to suitable process units. Such waste streams are referred to as process sources and all the process units that can accept recycled waste streams are named process sinks. Typically, fractions of process sources are mixed together with an external fresh source (*e.g.*, pure stream of the target compound being recycled) to satisfy the required amount of the target compound in a given process sink. At the same time, each process sink can accept a predefined maximum allowable level of impurity.

With this formulation (**Figure 1**), a linear optimization problem can be defined to maximize the amount of process sources recycled to process sinks and equivalently minimize the required amount of fresh resources:

$$\text{Maximize } W^{rec} = \sum_n w_n^{rec} \quad (\text{or Minimize } F = \sum_n f_n) \quad \text{where } n = (1, \dots, N_{sinks}) \quad (1)$$

s.t:

$$w_i = \sum_n w_{n,i}^{rec} + w_i^{waste} \quad \forall i = (1, \dots, N_{sources}) \quad (2)$$

$$g_n = \sum_i w_{n,i}^{rec} + f_n \quad \forall n = (1, \dots, N_{sinks}) \quad (3)$$

$$g_n z_n = \sum_i w_{n,i}^{rec} y_i \quad \forall n = (1, \dots, N_{sinks}) \quad (4)$$

$$0 \leq z_n \leq z_n^{max} \quad \forall n = (1, \dots, N_{sinks}) \quad (5)$$

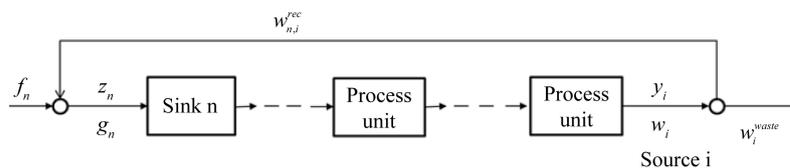
$$f_n \geq 0 \quad \forall n = (1, \dots, N_{sinks}) \quad (6)$$

$$w_{n,i}^{rec} \geq 0 \quad \forall i = (1, \dots, N_{sources}) \quad \text{and} \quad \forall n = (1, \dots, N_{sinks}) \quad (7)$$

$$w_i^{waste} \geq 0 \quad \forall i = (1, \dots, N_{sources}) \quad (8)$$

where  $w_n^{rec}$  is the amount recycled to process sink  $n$  (*i.e.*,  $w_n^{rec} = \sum_i w_{n,i}^{rec}$ ),  $f_n$  is the amount of fresh resource used in process sink  $n$ ,  $w_{n,i}^{rec}$  is the amount of process source  $i$  that is recycled to process sink  $n$  and  $w_i^{waste}$  is the amount of process source  $i$  that is not recycled and therefore is treated as waste,  $y_i$  is the impurity fraction of the process source  $i$ ,  $z_n$  is the impurity fraction at the process sink  $n$ ,  $z_n^{max}$  is the maximum allowable level of impurity of the process sink  $n$ ,  $w_i$  is the amount of waste of each process source,  $g_n$  is the required amount of the target compound at each process sink,  $W^{rec}$  and  $F$  are the total amounts of recycling and fresh resources used, respectively.

It should be noted that in this particular problem formulation there is one type of fresh source assumed to be 100% pure, all possible impurities are lumped into one, and there is no minimum allowable level of impurity at any process sink. Typically, the values of  $g_n$  and  $z_n^{max}$  at every process sink are known, as well as the waste



**Figure 1.** Process representation with sinks, intermediate units, sources and recycling (the symbols are defined in Equations (1) to (8)).

loads ( $W_i$ ) and their compositions ( $y_i$ ). Thus, solving the linear problem consists in finding the optimal values of  $w_{n,i}^{rec}$ .

This problem is based on the simplifying assumption that the impurities in the process sources recycled back to the process sinks after integration have a negligible impact on the respective process output variables, at least as long as the maximum allowable impurity constraints are satisfied; in this way, there is no need to include any process model that describes the impact of the recycled impurities on the performance of the process units. This assumption practically allows this kind of mass integration problem to be linear and solved to global optimality. Typically, there may be more than one optimal solutions, characterized by different connections among process sources and sinks.

## 2.2. Ordering Algorithm

Solving the direct recycling problem, as presented above, leads to an optimal target in the form of minimum fresh sources or maximum recycling of a target compound and one (of possible many or even infinite) optimum source-to-sink connectivity pattern that corresponds to the optimal target. In large problems (*i.e.*, with many sources and sinks), the structure of the global solution can be quite complicated, impractical to realize in its full extent or all at once, and difficult to interpret. Additionally, significant simplifications may have been made in the problem formulation and this may have an impact on the reliability of the overall solution of the direct recycling problem from industrial practice point of view.

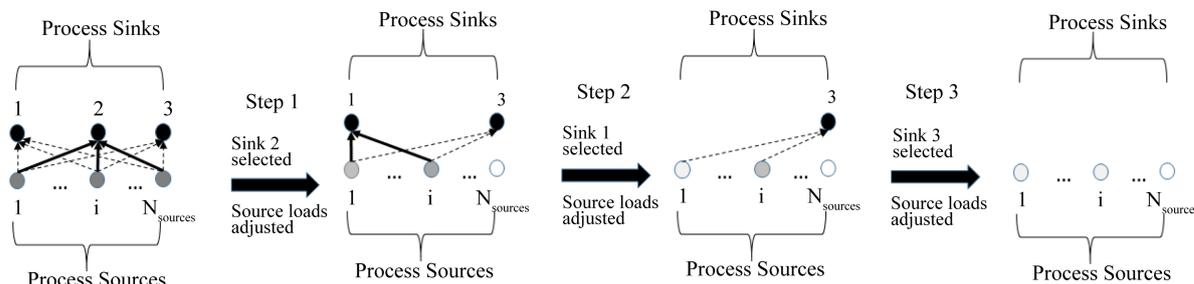
For these reasons, a sequential methodology implying some ordering of recycling actions can be beneficial. In principle, it would be desirable that a sequential method can also lead to the same overall optimum of the superstructure approach, if executed in all its steps. It is also important for such an approach to result in a decreasing gradient of the respective cumulative curve for the recycling loads as a function of the steps of the sequential method; in this way, the user can terminate the computational procedure quite in advance without having to compute many subsequent steps of marginal improvement. It is also important to know if the obtained optimum at any termination point is also globally optimum given the number of steps (*i.e.*, not only sequentially optimum in the sense that an optimum is found at any step given the decisions made at previous steps) or at least to have a list of necessary conditions for this to be true.

One can think of many ways to define the concept of sequential steps in the direct recycling problem. For instance, one step of the sequential method could be defined as one source-to-sink connection (*i.e.*, using the terminology of the direct recycling formulation this corresponds to the variable  $w_{n,i}^{rec}$ ). In this work, we present an ordering algorithm for the process sinks of the direct recycling problem, the order being based on the total recycled flows from all process sources to each sink ( $w_n^{rec} = \sum_i w_{n,i}^{rec}$ ). Thus, one step of the sequential method is defined as the selection of the optimal process sink to integrate with the available process sources. The sinks integrated in previous steps and part of the respective sources assigned to them are kept fixed (*i.e.*, they are eliminated from the rest of the sequential selection procedure). Therefore, at each step there is a remaining set of process sinks and process sources to which PGP is applied to find the optimum sink and process sources recycling waste loads to it. **Figure 2** presents schematically this sequential logic of the ordering algorithm for an example of three process sinks. **Figure 3(a)** and **Figure 3(b)** present in detail the steps of the ordering algorithm and the PGP applied to its step, respectively. The following remarks should be made with respect to the application of the ordering algorithm.

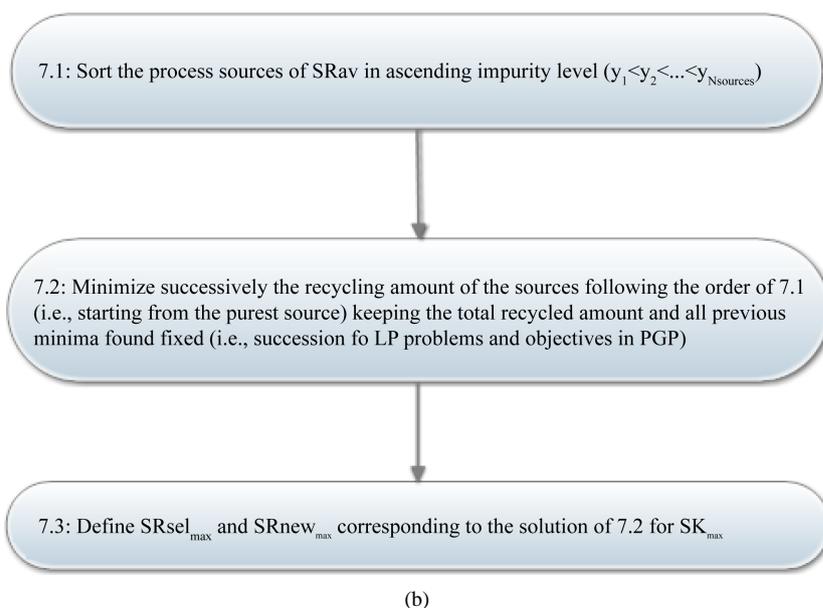
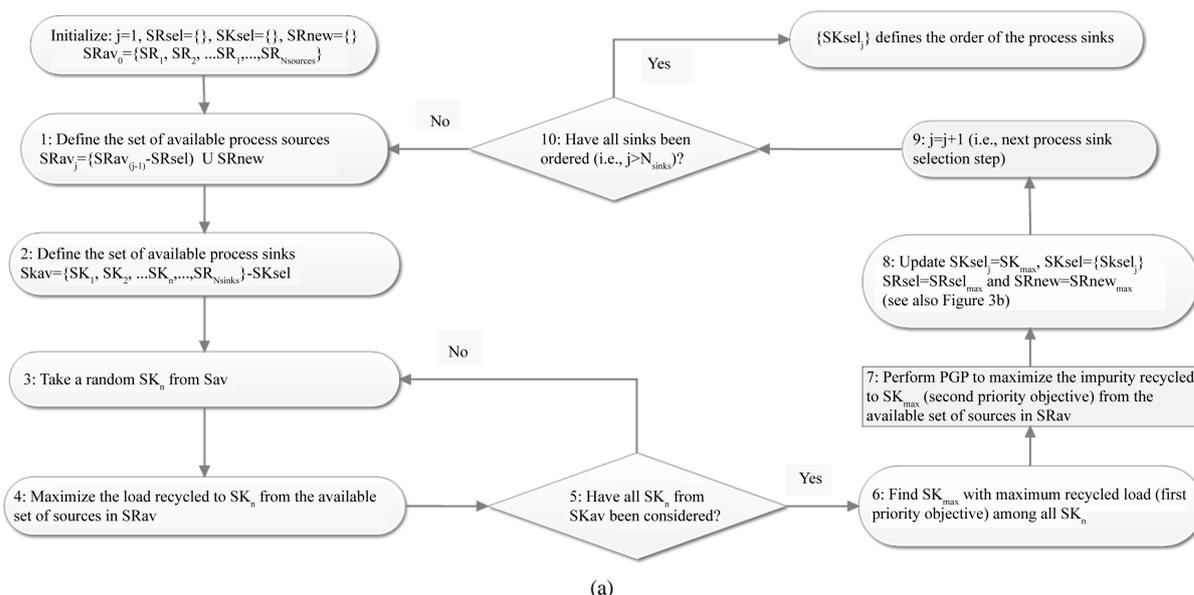
**Remark-1:** In **Figure 3(a)** the set of available process sources (SRav<sub>j</sub>) is updated at every process sink selection step ( $j$ ) according to the procedure of box-1 of the algorithmic scheme, which is illustrated in *example-1*.

### Example-1

At some process sink selection step  $j$  of the algorithm there are three available process sources, namely SR<sub>1</sub>



**Figure 2.** Schematic representation of the sequential logic of the ordering algorithm for an example of three process sinks. The gradual depletion of process sources is graphically shown as less intense gray shading.



**Figure 3.** (a) Sequential algorithm for ordering the process sinks in a direct recycling mass integration problem; (b) Pre-emptive goal programming (PGP) as an inherent step of the sequential ordering algorithm.

(50 kg/sec),  $SR_2$  (40 kg/sec), and  $SR_3$  (60 kg/sec), the values in parenthesis denoting the loads of the sources. This means that  $SRav_j = \{SR_1, SR_2, SR_3\}$ . After following the procedures described in box-2 to box-7 of **Figure 3(a)**, a process sink is selected ( $SK_{max}$ ) to which the process sources in  $SRav_j$  recycle the maximum possible load and impurity. For instance, this could be a process sink with  $g_n$  of 50 kg/sec, where according to the impurity concentrations in  $SR_1$ ,  $SR_2$ ,  $SR_3$  respectively and the maximum allowable impurity level of the sink, 20 kg/sec and 30 kg/sec are recycled from  $SR_1$  and  $SR_2$ , respectively, and no recycle from  $SR_3$ . This means that  $SRsel$  and  $SRnew$  in box-8 of **Figure 3(a)** can now be defined in the following way:  $SRsel = \{SR_1, SR_2\}$  and  $SRnew = \{SRnew_1, SRnew_2\}$  where  $SRnew_1$  and  $SRnew_2$  have now remaining loads of 30 kg/sec and 10 kg/sec, respectively. Moving now to the next step ( $j = j + 1$ ) and updating the available process sources set in box-1 of **Figure 3(a)**,  $SRav_{j+1} = (\{SR_1, SR_2, SR_3\} - SRsel) \cup SRnew = \{SR_3\} \cup SRnew = \{SRnew_1, SRnew_2, SR_3\}$ .

*Remark-2:* The impurity concentrations in the process sources are assumed not to change after closing the recycle loops. However, this is not a particularity of the sequential ordering algorithm presented here, since the same assumption is implicitly made in the general formulation of direct recycling problems (*i.e.*, in its LP form), when it is stated that the recycled impurities do not affect the performance of the process sinks and, consequently, the composition of the process sources does not change.

*Remark-3:* As shown in box-8 of **Figure 3(a)**,  $SKsel$  is a sequence (*i.e.*, a set with ordered elements) denoted as  $SKsel = \{SKsel_j\}$ . Thus,  $SKsel$  includes the ordered sinks and at the same time performs the reduction of the set of available process sinks  $SKav_j$  for every process sink selection step (*i.e.*, box-2 of **Figure 3(a)**).

*Remark-4:* From box-3 to box-5 there is an internal loop at every process sink selection step. In this loop the maximum recycled amount to each process sink in  $SKav_j$  is independently calculated (*i.e.*, by solving a LP such as the one described by Equations (1) to (8) with  $N_{sinks} = 1$ ) considering the process sources in  $SRav_j$ . It should be reminded here that the solution of each LP problem may not be unique, since more than one (or even infinite) combinations of the process sources may be maximizing the recycled amount ( $w_n^{rec}$ ) to a given process sink  $SK_n$ .

*Remark-5:* After finding  $SK_{max}$  (box-6 of **Figure 3(a)**) the PGP procedure follows (box-7 of **Figure 3(a)**), which is presented in **Figure 3(b)**. In this, the level of impurity is maximized or, equivalently, the recycled amount of the purest sources is minimized (box 7.2 of **Figure 3(b)**) by replacing the purer recycled streams with more impure ones. This is realized through a succession of linear problems in a PGP formulation, namely keeping constant each objective found in one LP as an additional constraint for the next LP in succession.

*Remark-6:* In case that in a process sink selection sept  $j$ , more than one  $SK_{max}$  are found in box-6 of **Figure 3(a)**, they all undergo the procedure of box-7 of **Figure 3(a)**, and the sink with the highest flow of recycled impurity has the precedence and is selected in  $SKsel$ .

*Remark-7:* Since the availability of process sources at  $SRav_j$  is more limited than at  $SRav_{j-1}$ , it follows that  $SKmax_j \leq SKmax_{j-1}$ . This means that the slope of the cumulative curve representing the total recycled amount at each process selection step will be decreasing.

*Remark-8:* At the end of the ordering algorithm, when all the sinks have been selected for recycling, the total amount recovered from the waste streams corresponds necessarily to the one calculated in the general LP problem (Equations (1) to (8) considering all the sinks simultaneously). Actually, it is already known that starting from any sink and following a similar procedure like the one described in **Figure 3(b)**, a unique target for the direct recycling can be reached. The procedure presented here is simply a specific, not random, selection of the process sinks, which can, therefore, be considered as a subcase of the random sink selection proposed by El-Halwagi [6]. Of course, the ordering algorithm can never find a superior total amount recycled, since it is known that the general LP formulation can be solved to optimality.

In the same way, a subgroup of process sinks is characterized by a fixed target for maximum recycle, given a set of available process sources. Therefore, at a general step of the process sink ordering algorithm, the total amount of recycle corresponds necessarily to the target quantities calculated in the general LP problem (Equations (1)-(8)) considering that group of  $n$  process sinks. The ordering algorithm is further illustrated in example-2.

### Example-2

The sets of process sinks and process sources in **Table 1** are considered to realize mass integration through direct recycling. These designate the initial  $SKav$  and  $SRav_1$  sets, respectively. If no mass integration was performed, the process sinks would require 280 kg/sec of the fresh target compound, while 310 kg/sec of waste

**Table 1.** Process sinks and process sources of example-2.

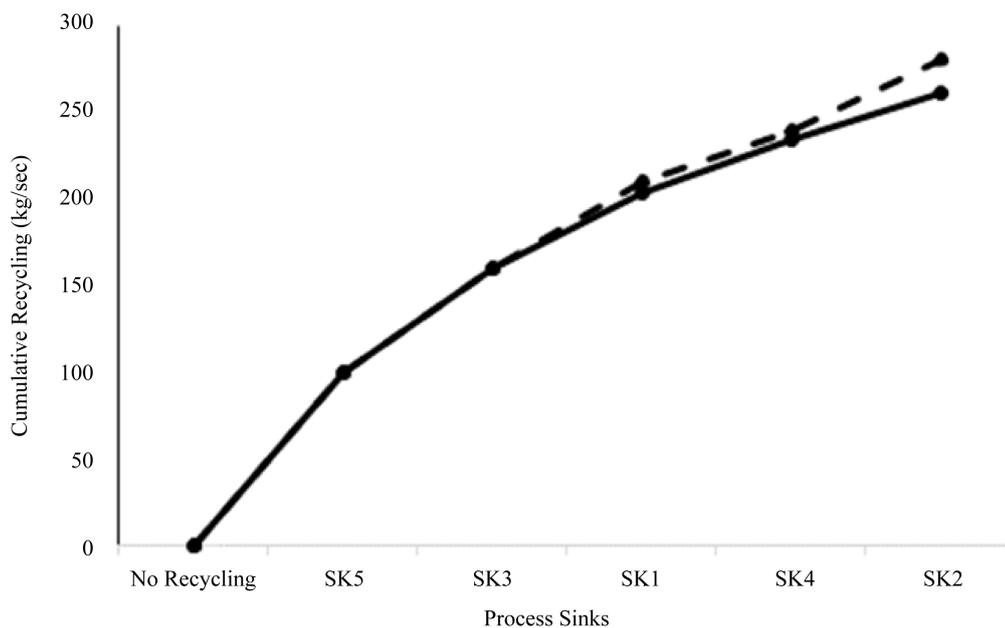
	Process sinks		
	Required amount of target compound (kg/sec)	Maximum allowable impurity (% w/w)	Maximum allowable impurity (kg/sec)
SK1	50	2	1
SK2	40	5	2
SK3	60	10	6
SK4	30	20	6
SK5	100	25	25
	Process sources		
	Waste load kg/sec	Impurity content (% w/w)	Impurity content (kg/sec)
SR1	30	1	0.3
SR2	40	5	2
SR3	50	10	5
SR4	50	15	7.5
SR5	40	20	8
SR6	100	30	30

should be disposed.

In this case, for the first process selection step ( $j = 1$ ), one may heuristically start from SK5 in box-3, namely the sink with the maximum required amount of target compound; if this amount could be recycled from the available process sources, there would be clearly no further need to check the rest of the sinks at this process selection step. One solution of the respective LP maximization problem could be to recycle the waste loads of SR3 and SR4. This would satisfy recycling 100 kg/sec to SK5, but only 12.5 kg/sec of impurities would be recycled, while SK5 can take up to 25 kg/sec of impurities. According to the PGP problem in box-7, the goal is to successively replace purer with more impure sources to achieve the goal of maximizing the impurity (or equivalently minimizing the use the purer sources at earlier steps of the process sink selection). It can be easily verified that solving the respective PGP problem results in using SR4 (6.7 kg/sec), SR5 (40 kg/sec) and SR6 (53.3 kg/sec). This recycles 100 kg/sec to SK5 and 25 kg/sec of impurity, which is its maximum allowable impurity level. Thus, in box-8 of **Figure 3(a)**,  $SR_{sel} = \{SR4, SR5, SR6\}$ ,  $SR_{new} = \{SR4_{new}, SR5_{new}, SR6_{new}\}$ , where  $SR4_{new} = 43.3$  kg/sec,  $SR5_{new} = 0$  kg/sec and  $SR6 = 47.7$  kg/sec, while  $SK_{sel} = \{SK_{sel_1}\} = \{SK5\}$ . Finally, making the first algorithmic loop to define the new available set of sources (box-1 of **Figure 3(a)**), it leads to  $SR_{av_2} = \{SR1, SR2, SR3, SR4_{new}, SR6_{new}\}$ ,  $SR5_{new}$  being eliminated since all of it has been recycled in this process sink selection step.

Using the same heuristic like before for the next process selection step ( $j = 2$ ), it now makes sense to start from SK3 in box-3. Doing so, one can easily find that the maximum required amount of target compound (60 kg/sec) can be indeed recycled to SK3 and the impurities can be maximized and reach again the maximum allowable impurity level with the following solution: SR2 (5 kg/sec), SR3 (50 kg/sec) and  $SR4_{new}$  (5 kg/sec). Thus, in box-8 of **Figure 3(a)**,  $SR_{sel} = \{SR2, SR3, SR4_{new}\}$ ,  $SR_{new} = \{SR2_{new}, SR3_{new}, SR4_{new}\}$ , where  $SR2_{new} = 35$  kg/sec,  $SR3_{new} = 0$  kg/sec and  $SR4_{new} = 38.3$  kg/sec, while  $SK_{sel} = \{SK_{sel_2}\} = \{SK5, SK3\}$ . Then, making the second algorithmic loop to define the new available set of sources (box-1 of **Figure 3(a)**), it leads to  $SR_{av_3} = \{SR1, SR2_{new}, SR4_{new}, SR6_{new}\}$ ,  $SR3_{new}$  being eliminated since all of it has been recycled in this step of the process sink selection.

Following this procedure until all sinks are ordered ( $j = 5$ , not presented here because of space limitations) results in the following ordering of process sinks  $SK_{sel} = \{SK_{sel_3}\} = \{SK5, SK3, SK1, SK4, SK2\}$ . The respective cumulative curve of the recycled amounts is presented in **Figure 4**. As it can be seen, 261.3 kg/sec are



**Figure 4.** Cumulative recycling curve (solid line) according to the sequential ordering algorithm applied to example-2. The dashed line represents the cumulative curve of the required amount of the target compound for the same order of the process sinks.

recycled from the process sources to the process sinks, reducing the process requirements for fresh target compound to only 18.7 kg/sec and the respective waste streams to 48.7 kg/sec. It can be easily verified that the same targets are calculated by the general LP problem of Equations (1) to (8), when all process sinks are considered simultaneously.

Comparing the recycling cumulative curve with the cumulative curve of the required amount of target compound, it can be seen that no fresh compound is required in process sinks SK5 and SK3. It can also be seen that the cumulative recycling curve has a decreasing slope, which, however, in this case does not lead to marginally increasing recycling steps, and, therefore, it would not make sense to terminate the algorithm before considering all process sinks.

### 2.3. Analysis of the Optimality of the Process Sinks Ordering

The ordering algorithm assures the conditional optimality of the process sink selection at a given algorithmic step (*i.e.*, given the previously selected process sinks) and also that the recycled amount to it cannot be superior to any previous one with the available set of process sources. However, it does not guarantee that for a given cardinality of the process sink selection (*i.e.*, ordering the first  $n$  sinks out of the totality of  $N_{sinks}$ ) the sequential procedure leads to the maximum cumulative recycling among any other group of  $n$  sinks. Nevertheless, one trivial condition can point out whether another group of  $n$  sinks has the potential to lead to a higher recycled amount or, instead, the first  $n$  sinks of the ordering algorithm are globally optimal. The condition requires that the cumulative recycling to the first  $n$  ordered process sinks is superior to the cumulative required amount of the target compound of any other group of  $n$  sinks. This is expressed by Equation (9):

$$R_1 + R_{2|1} + \dots + R_{n|1,2,\dots,(n-1)} \geq G(SK_{k_1}) + G(SK_{k_2}) + \dots + G(SK_{k_n}) \quad (9)$$

$$\forall \{SK_{k_1}, SK_{k_2}, \dots, SK_{k_n}\} \neq \{SK_1, SK_2, \dots, SK_n\}$$

where  $R_{n|1,\dots,(n-1)}$  refers to the recycled amount to process sink  $n$  ( $w_n^{rec}$ ) given the  $n - 1$  previously ordered process sinks and  $G(SK_{kn})$  refers to the required amount of the target compound at a process sink  $kn$  (*i.e.*, with respect to the notation in Equations (1) to (8),  $G(SK_{kn}) = g_{kn}$ ).

Thus, we can make the following statements:

*Statement-1:* Any group of process sinks that violates condition (9) may be characterized by a higher target

for maximum recycling (*i.e.*, such a group is a candidate for outperforming the ordered group of process sinks and thus it should be further tested, for example, by solving the respective linear optimization problem, expressed by Equations (1) to (8), for this specific candidate group of process sinks.

*Statement-2:* If no group violates the condition (9), then the ordering algorithm has identified the globally maximum recycling for the cardinality of sinks at the specific step of the process sink selection. Therefore, the condition (9) is a necessary condition.

For instance, in the previously presented example-2, at the third process sink selection step, where  $SK_{sel} = \{SK_5, SK_3, SK_1\}$ , the condition (9) takes the form:

$$R_5 + R_{3|5} + R_{1|5,3} > G(SK_{k1}) + G(SK_{k2}) + G(SK_{k3}), \forall \{SK_{k1}, SK_{k2}, SK_{k3}\} \neq \{SK_5, SK_3, SK_1\}. \quad (10)$$

In this particular example,  $R_5 + R_{3|5} + R_{1|5,3} = 204$  kg/sec and no other triplet of process sinks exists that violates the condition (10). Thus, at this process sink selection step, the sequential algorithm demonstrates not only conditional but also global optimality.

### 3. Case Studies and Results

#### 3.1. Case Study-1: Statistical Evaluation of the Performance of the Ordering Algorithm

Since the ordering algorithm does not always guarantee global optimality for a given cardinality of ordered process sinks (*i.e.*, other than the first and the last step of the algorithm, namely when only the first sink or all the sinks of the direct recycling problem are ordered), it is interesting to test the frequency of such cases under different conditions. Thus, a thorough computational screening of direct recycling problems was performed including various factors differentiating the scenarios tested. **Table 2** summarizes these factors and their levels of differentiation defining the scenarios tested.

The first factor refers to the target compound for recycling with three levels of differentiation, namely inert, generated or consumed compounds in the process system. The tested levels of depletion or excess of generation for the last two cases range from 10% to 90% with respect to the fresh source required before any recycling. The second factor refers to cases with either higher number of sources or sinks, the number of sources and sinks ranging each from 1 to 15. The third factor refers to the type of sampling with respect to the loads and the impurity concentrations of the process sources and the required amount of target compound and maximum allowable level of impurity of the process sinks. Three types of sampling were considered: latin-hypercube without space filling, allowing repetitions and totally randomized.

Combinations of these factors have been used to define the different scenario categories (e.g., consumed compound with 90% depletion, higher number of process sources than sinks and totally randomized sampling, may refer to one combination of factors defining one scenario). Multiple samples were created (*i.e.*, 1000 for each sub-problem) based on the size of the problem (*i.e.*, number of sinks) and the loads and impurity combinations.

For the evaluation of the scenarios, three indices were defined (Equations (11)-(13)):

$$Ind1 = N_v / N_{sinks} \quad (11)$$

$$Ind2 = N_h / N_{sinks} \quad (12)$$

$$Ind3 = Ind2 / Ind1 \quad (13)$$

**Table 2.** Factors considered in creating diverse direct recycling scenarios (in parenthesis the abbreviated name is given for every level of differentiation).

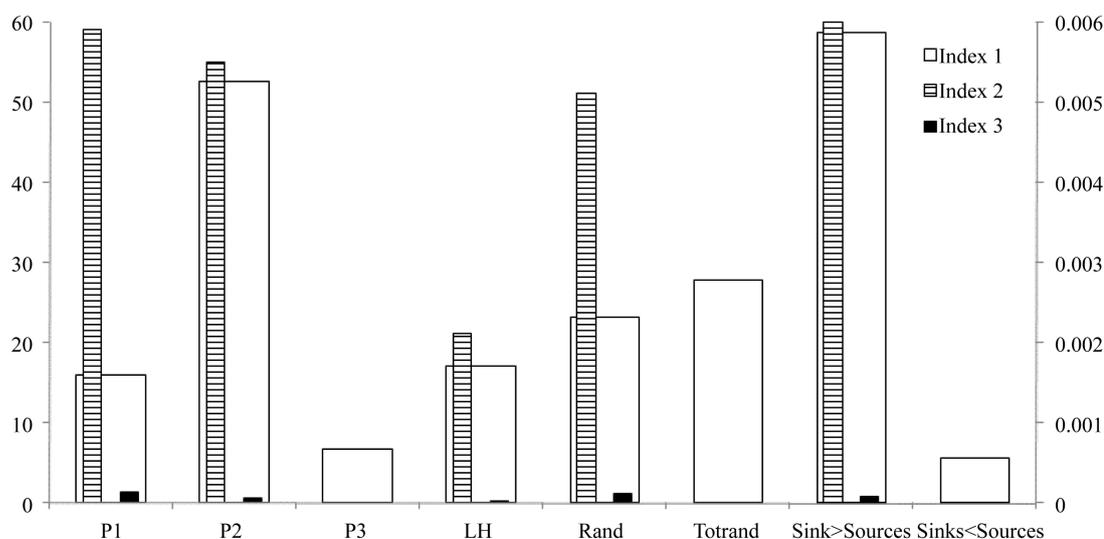
Factors of differentiation	Levels of differentiation		
Type of target compound to be recycled	Inert (P1)	Depleted (P2) 10 to 90%	Generated (P3) 10 to 90%
Number of process sources and sinks	Sources > sinks		Sinks < sources
Type of sampling	Latin hypercube (LH)	Allowing repetitions (Rand)	Totally random (Totrand)

where  $N_v$  is the number of cases where condition (9) was violated,  $N_h$  refers to those cases among  $N_v$  which reach a higher recycling than the one proposed by the sequential ordering algorithm for a given process sink selection step, and  $N_{sinks}$  is the number of sinks of the specific scenario tested.

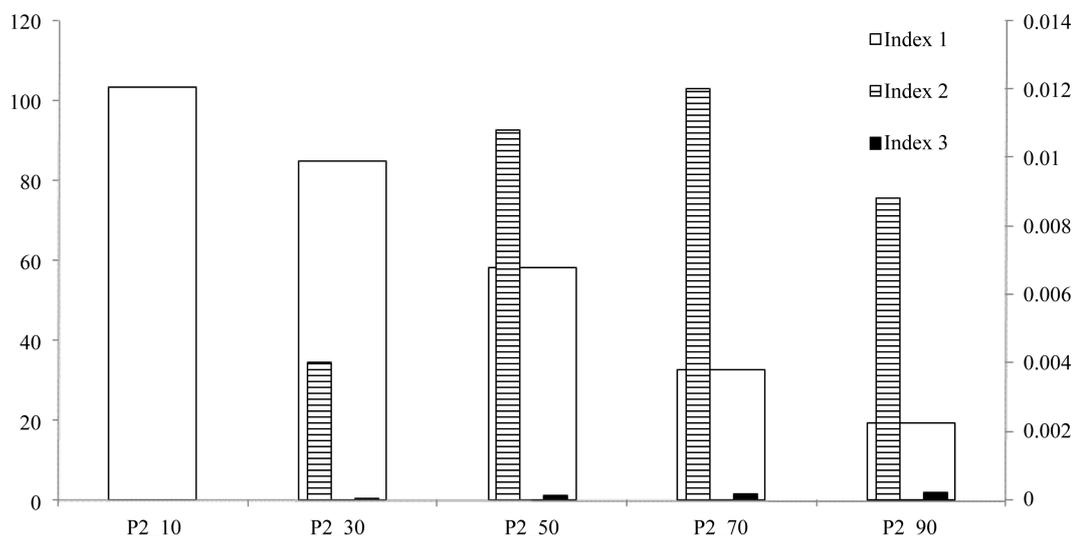
The number of sinks obviously defines the steps of the process sink selection and, thus, also the potential violation points for condition (9). Therefore,  $Ind1$  and  $Ind2$  are normalized to  $N_{sinks}$  so as to compare lumped cases with different number of sinks (e.g., all cases with more sources than sinks as one “population” for the statistical analysis) through aggregated statistical metrics (e.g., mean values, standard deviations etc.). Since  $N_{sinks}$  refers also to the number of sequential steps of the ordering algorithm,  $Ind1$  values around 1, for instance, means that one group of sinks per process sink selection step, on average, is violating the condition (9). Similarly,  $Ind2$  values around 1 means that one group of sinks per process sink selection step, on average, recycles more than what is proposed by the sequential ordering algorithm and, thus,  $Ind3$  values close to 1 means that almost all of the cases that violated the condition (9) achieve, indeed, a higher recycling than what is proposed by the sequential ordering algorithm.

The results of the statistical evaluation on the basis of the three indices and the lumped cases of **Table 2** are presented in **Figure 5**. For instance, P1 refers to lumping all scenarios where the target compound resembles the behavior of an inert compound, namely the total amount of the compound in the process sources is equal to the total amount required in the process sinks. This means that this case includes scenarios with all three different types of sampling, more sources than sinks and *vice versa*. From the results in **Figure 5**, it can be inferred that a significant number of groups of process sinks for each process sink selection step violates the condition (9), since the values of  $Ind1$  are significantly higher than 1. This is more evident in the cases of depleting target compounds (*i.e.*, P2) and more process sinks than process sources (Sinks > Sources). Nevertheless, the values of the other two indices are extremely low. For instance, according to the values of  $Ind2$ , the number of groups of process sinks with superior performance compared to the respective ordered groups of process sinks of the sequential algorithm are two to three orders of magnitude lower than the process sink selection steps (*i.e.*,  $Ind2$  ranges between 0.002 and 0.006). Similarly, according to the values of  $Ind3$ , on average only 0.01% of the groups violating the condition (9) outperform the group of sinks selected by the sequential ordering algorithm at each selection step. This could also indicate that stricter conditions than the condition (9) should be identified for a more efficient screening of those groups of process sinks with the potential to outperform the ordered groups of process sinks with the same cardinality. If necessary and sufficient conditions can be identified, then obviously indices such as  $Ind2$  and  $Ind3$  will be redundant.

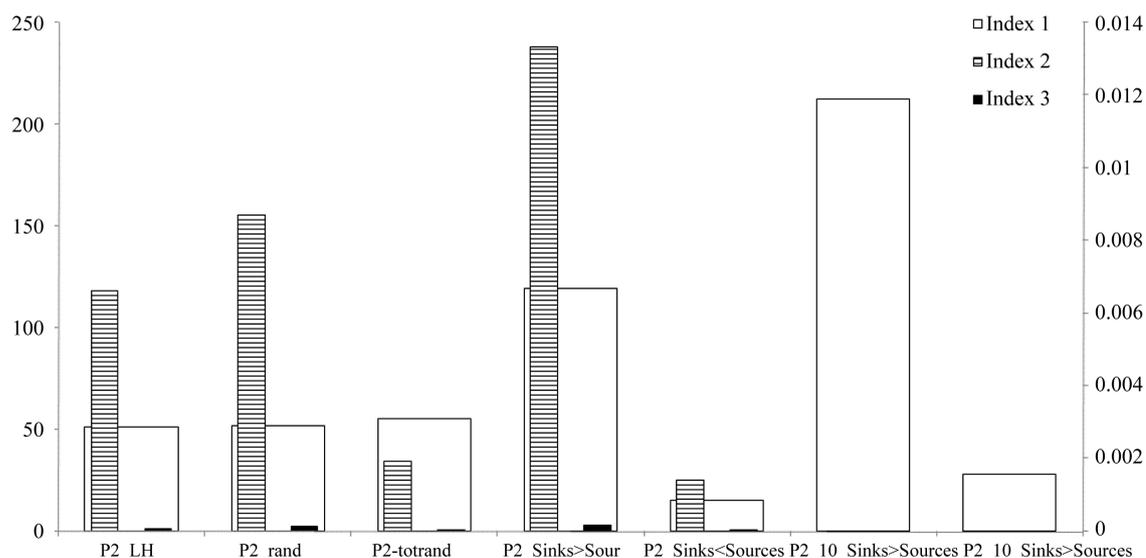
A more detailed look into sub-cases of the lumped P2 case is provided in **Figure 6** and **Figure 7**. More specifically, in **Figure 6** the impact of the level of depletion is analyzed for the scenarios in the P2 case, which was one of the cases with relatively high values for  $Ind1$  and  $Ind2$ . It is obvious that there are two opposite



**Figure 5.** Trimmed mean values (5% - 95% interval) of  $Ind1$  (left y-axis),  $Ind2$  and  $Ind3$  (right y-axis) for the lumped categories of **Table 2**.



**Figure 6.** Trimmed mean values (5% - 95% interval) of *Ind1* (left y-axis), *Ind2* and *Ind3* (right y-axis) for sub-cases with respect to different depletion levels for the P2 lumped category of **Table 2**. The numerical index refers to the level of remaining compound in the process sources with respect to the required amount in the process sinks (e.g., P2\_10 means that 10% of the required amount of the target compound in the process sinks is found in the process sources, the rest 90% being depleted in the intermediate process units).



**Figure 7.** Trimmed mean values (5% - 95% interval) of *Ind1* (left y-axis), *Ind2* and *Ind3* (right y-axis) for various sub-cases of the P2 lumped case of **Table 2**.

trends as the level of depletion is decreasing (*i.e.*, P2\_10 means that only 10% of the required amount of the target compound in the process sinks exists in the process sources, while the rest 90% is depleted in the intermediate process units): the values of *Ind1* generally decrease while the values of *Ind2* and thus also of *Ind3* increase. Comparing these trends with the P1 and P3 values of the indices in **Figure 5**, it can be inferred that the more balanced the amounts of the targeted compound for recycling are between process sinks and process sources, the higher the chance (although in absolute numbers still very low) to identify groups of process sinks outperforming those of the sequential ordering algorithm with the same cardinality. In the case of imbalance between these amounts, the case of depleted target compound (P2) is more sensitive with respect to the existence of such groups.

Other sub-cases of P2, such as those presented in **Figure 7**, do not appear to significantly reduce the *Ind1* values, while *Ind2* and *Ind3* values remain generally very low. For the *Ind2* and *Ind3* values, the impact of a high depletion level of the target compound is dominant compared to other factors; this is exemplified by the comparison between the cases P2\_Sinks > Sources and P2\_10\_Sinks > Sources, considering also the values for P2\_10 presented in **Figure 6**. A detailed list of the statistical evaluation for all the scenarios tested is provided in ESI (Part 2).

### 3.2. Case Study-2: Gradual Process Retrofitting

One particular case of interest for the application of the sequential ordering algorithm is when the realization of mass integration through direct recycling can be carried out only by a gradual retrofitting, namely reducing the fresh source of the target compound to one process sink at a time. In this case study, a pharmaceutical process producing an intermediate has to undergo mass integration by recycling of its most relevant solvent, the N-methylpyrrolidone, whose price is 2500 \$/tonne. **Table 3** provides the respective sets of the process sources and sinks.

The recycling of the process sources to the sinks can be realized without serious problems that would interrupt the production of the main chemical, if one sink is disconnected from the rest of the process at a time, keeping all the other sinks in normal operation. Before any process sink is disconnected from the rest of the process, it discharges enough amount to a buffer tank to keep the process operating with the same load discharged into the process sources. Then, it is linked with the pipes of the process sources to receive the recycling load and, subsequently, reconnected with the rest of the process. The buffer tank is then used for the same purpose in the next process sink, and so on. Disconnection of the sink and reconnection with the process sources requires in total 3 days. This means that realizing any direct recycling strategy for all 5 process sinks requires 15 days.

In this kind of problem, the sequential ordering algorithm can provide the optimal order of the sinks so that the sinks with the highest recycling are disconnected and reconnected first. In this way, fresh resources are saved during the intermediate period of realizing the mass integration strategy, although at the end of the procedure other sink orders would also recycle the same total amount. Therefore, it is convenient to integrate the process sinks following the cumulative curve to obtain the highest economic saving from the first step.

The application of the algorithm results in the process sink ordering presented in **Table 4**, while **Table 5**

**Table 3.** Process sinks and process sources of the case study-2.

	Process sinks		
	Required amount of target compound (kg/hr)	Maximum allowable impurity (% w/w)	Maximum allowable impurity (kg/hr)
SK1	300	1	3
SK2	250	2	5
SK3	280	10	28
SK4	150	20	30
SK5	100	30	30
	Process sources		
	Waste load (kg/hr)	Impurity content (% w/w)	Impurity content (kg/hr)
SR1	100	5	5
SR2	120	10	12
SR3	180	20	36
SR4	120	30	36
SR5	150	50	75
SR6	150	70	105

**Table 4.** Ordered process sinks according to the sequential ordering algorithm for the case study-2.

Ordered sinks	Recycled amount per sink (kg/hr)	Cumulative recycled amount (kg/hr)	Cumulative solvent savings (\$)	Sources connected to sink (kg/hr)					
				SR1	SR2	SR3	SR4	SR5	SR6
SK3	275	275	49500	100	120	55	0	0	0
SK4	141.7	416.7	124506	0	0	125	16.7	0	0
SK5	100	516.7	217512	0	0	0	100	0	0
SK2	11.3	528	312552	0	0	0	3.3	8	0
SK1	6	534	408672	0	0	0	0	6	0

**Table 5.** Alternative ordering of process sinks (*i.e.*, not based on the sequential ordering algorithm) for the case study-2.

Ordered sinks	Recycled amount per sink (kg/hr)	Cumulative recycled amount (kg/hr)	Cumulative solvent savings (\$)	Sources connected to sink (kg/hr)					
				SR1	SR2	SR3	SR4	SR5	SR6
SK1	60	60	10800	60	0	0	0	0	0
SK3	230	290	63000	40	120	70	0	0	0
SK2	25	315	119700	0	0	25	0	0	0
SK4	128.3	443.3	199494	0	0	85	43.3	0	0
SK5	90.7	534	295614	0	0	0	76.7	14	0

presents an alternative solution (*i.e.*, not obtained by the sequential ordering algorithm), resulting in the same overall recycled amount. It is worth noticing that, although the two solutions have different structure (*i.e.*, source-to-sink connectivity), the same total amount is recycled from each process source. However, these two solutions have significantly different economic performance. The solution based on the sequential ordering algorithm saves 113058 \$ more in this intermediate period for realizing the full source-to-sink connectivity of the direct recycling solution. The reason for the better performance of the sequential ordering algorithm lies in the property of the monotonically decreasing gradient of its respective cumulative curve of the recycled amount.

### 3.3. Case Study-3: Direct Recycling in Processes Considering Their Nonlinear Behavior

Another possible application of the methodology is when the hypothesis of negligible impact on the process sink performance by the recycled impurities is not anymore valid. From modelling point of view, this means that a detailed model of the process sink is available, which can describe the impact of the impurities on the process sink output streams, this impact being nonlinear in the general case. In such cases, the calculation of the target for mass integration through direct recycling requires to solve a NLP optimization problem, whose solution can be hard to calculate. Besides the computational difficulties, another practical issue is related to the frequent use of advanced process simulators for rigorous modelling of the process units, which are not typically designed to handle NLP superstructure optimization. Moreover, in industrial practice, it is often required to verify and understand the impact of any process change (e.g., such as recycling impure process streams), even when rigorous process models are available.

To tackle this problem, the sequential ordering algorithm offers an interesting alternative; the optimal ordering of process sinks obtained by solving the respective linear representation of the problem can be considered as the basis for a subsequent sequential procedure to calculate the optimum recycling target of the nonlinear representation of the process. Thus, the steps of this new sequential procedure follow those of the ordering algorithm applied to the linear representation of the problem. However, because of the nonlinear relations among process streams and units, the impurity content of the process sources may change significantly when the recycling is realized; this in turn can cause the violation of the maximum allowable impurity constraints of the process sinks, if the recycling loads are not properly adjusted. This adjustment of the recycling loads is performed sequentially according to the order of the process sinks. This has the advantage of studying the nonlinear performance of on-

ly one sink at a time, starting from the sinks with the highest recycling loads, which are more likely to have a greater impact on the overall process performance.

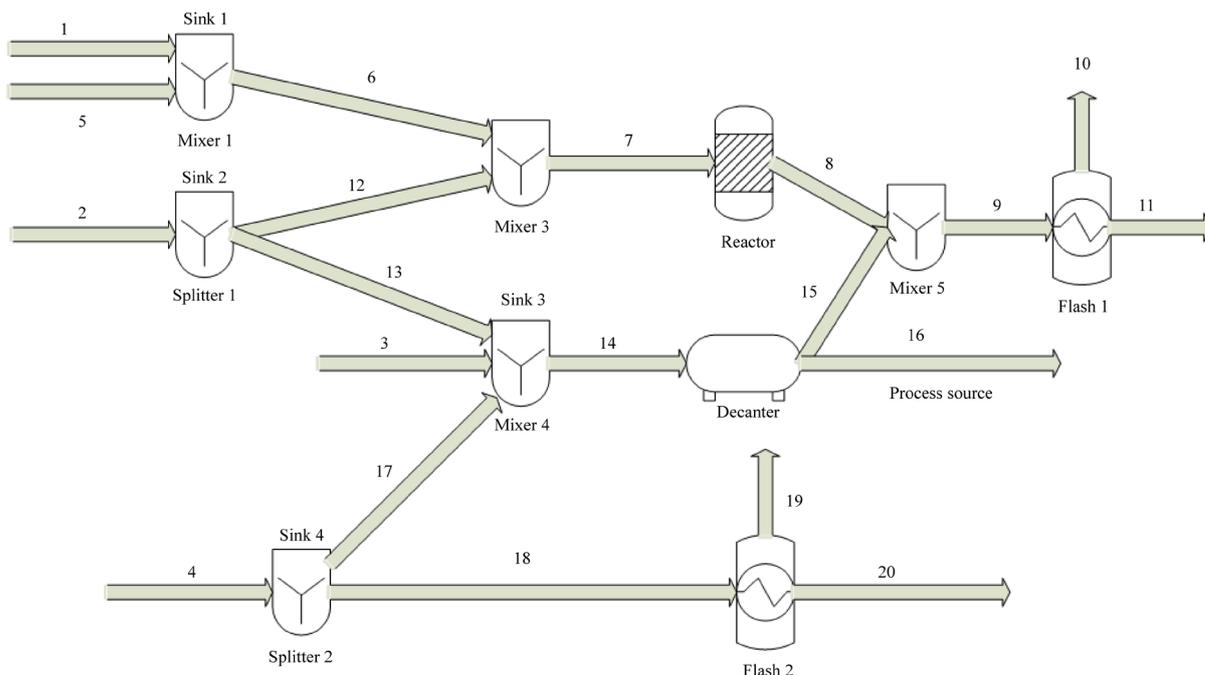
As an example, the process depicted in **Figure 8** is considered for treating compound B. In the continuously stirred 1 m<sup>3</sup> reactor, the following reaction takes place at 150°C and 1 bar:



The kinetics follow the power law with respect to compound concentrations, with Arrhenius expressions for the kinetic constants both for the direct and reverse reactions. The evaporation at flash-1 takes place at 150°C and 1 bar, the decanter operates at 50°C and 1 bar, and the evaporation at flash-2 at 70°C and 1 bar. The compositions of the streams for both the vapor-liquid and liquid-liquid phase separation at thermodynamic equilibrium are calculated with the Non-Random-Two-Liquid (NRTL) method. These kinetic expressions and thermodynamic models introduce strong nonlinearities in the performance of the respective process units. A full stream table of the process is provided in ESI (Part 3).

In this case study, it is desired to minimize the use of the fresh amount of compound A in the process sinks 1, 2, 3 and 4 through the recycling of the process stream 16 (“process source”). The relevant process sinks and source data are provided in **Table 6**.

First, the recycling target according to the linear representation of the problem is calculated resulting in 7.11 kmol/hr. The respective cumulative curve is presented in **Figure 9**. Subsequently, the connections between the process source and the sinks are sequentially realized in a process simulator capturing the nonlinear process behavior, starting from the first ordered sink. Each time, the recycled flow is reduced in case of violation of the maximum allowable level of impurity of the respective sink. The global target found in this way is 6.95 kmol/hr and the respective cumulative curve is presented in **Figure 9**. Solving the respective overall NLP superstructure problem (*i.e.*, with the help of a Sequential Quadratic Programming (SQP) procedure embedded in the process simulator) results in 6.97 kmol/hr and the equivalent cumulative curve is presented in **Figure 9**. In this case, the sequential procedure has provided a recycled target that is approximately 0.3% inferior compared to the rigorous solution (*i.e.*, the sequential ordering algorithm adapted to the nonlinear problem formulation has an almost identical cumulative curve compared to the equivalent curve of the SQP solution). They both recycle approximately 2% lower than the solution obtained by the linear problem formulation. Both deviations are only indicative in a rather simple direct recycling problem. It is a matter of our on-going research to test the performance of the adapted sequential ordering algorithm in more complicated nonlinear problems with significantly more



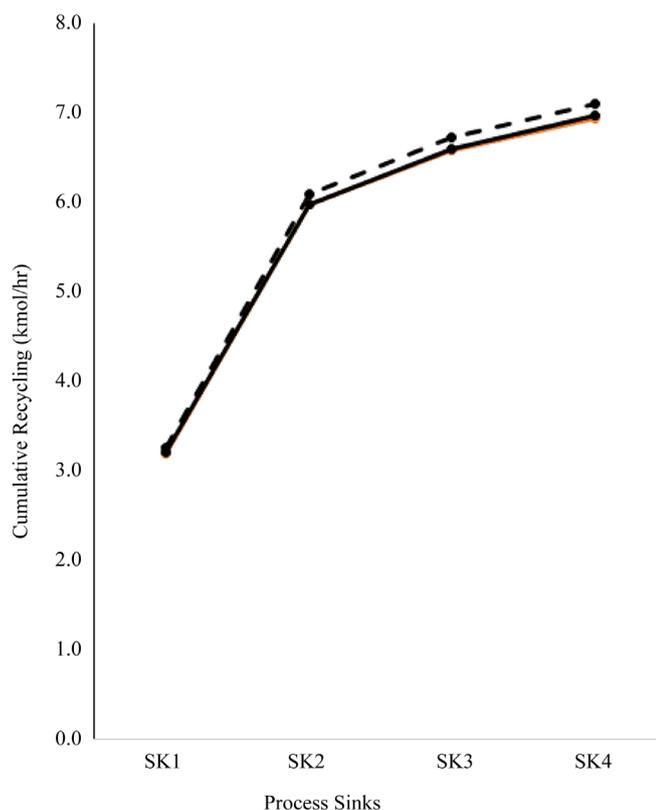
**Figure 8.** Process flowsheet of the case study-3.

**Table 6.** Process sinks and process sources of the case study-3.

	Process sinks		
	Required amount of target compound (kmol/hr)	Maximum allowable impurity (% mol/mol)	Maximum allowable impurity (kmol/hr)
SK1	13	20	2.6
SK2	15	15	2.25
SK3	10	5	0.5
SK4	10	3	0.3

	Process sources		
	Waste load (kmol/hr)	Impurity content (% mol/mol)	Impurity content (kmol/hr)
SR1	100	79.5	79.5



**Figure 9.** Cumulative curves based on the sequential ordering algorithm and a standard SQP superstructure solution for the case study-3. The dashed line refers to the linear representation of the problem, the black solid line to the nonlinear representation of the problem and the red solid line to the SQP superstructure solution.

process sources and sinks. This case study is only used to highlight a potential way to decompose a complicated NLP direct recycling problem with the help of the proposed sequential ordering algorithm for the process sinks.

#### 4. Conclusions

The design of optimal mass integration networks often results in complicated superstructure formulations and solutions, which can be cumbersome to interpret and realize in practice. Therefore, in many cases a more struc-

tured construction of the solution based on sequential approaches is advantageous. This work addresses this problem by proposing an algorithm for optimal ordering of the process sinks in direct recycling problems, which is compatible with the typical mass integration formulation and reaches the same recycling target in the case of the linear representation of the problem. The obtained order consists in selecting the optimal sink at a specific integration step given the selection of the previous steps and the remaining process sources. Such order is identified through a succession of preemptive goal programming problems, namely of optimization problems characterized by more objectives at different priority levels. Indeed, the target for each sink is obtained by maximizing the total flow recycled from the available process sources to this sink and then minimizing the use of pure sources, starting from the purest one; the hierarchy is respected through a succession of linear optimization problems with a single objective function.

In this study, the sequential ordering algorithm has undergone a thorough statistical test (case study-1) to identify the frequency of conditions under which, at a certain step, the algorithm has selected the group of sinks with the highest target for recycling (global optimality) among all the groups with the same cardinality. It has been shown that in the vast majority of the cases the algorithm identifies a globally optimum group of process sinks for each step of process sink selection: the most challenging cases are those where the process sinks are more than the process sources in inert or slightly decreasing target compound scenarios. Of course, the conditional optimality is always guaranteed at each process selection step, while ensuring that at the end of the process sink ordering procedure the maximum amount of the target compound is recycled. It would be, however, useful to define simple, yet stricter algebraic conditions than those proposed herein, to efficiently identify these groups of process sinks that need to be tested for outperforming the process sinks of the same cardinality selected by the sequential ordering algorithm.

Two more case studies were conducted to demonstrate the usefulness of ordering the process sinks. In the case study of gradual retrofitting (case study-2), ordering the process sinks resulted in significant economic savings, compared to a solution with an equivalent recycling target but with different ordering strategy. In the case study considering the nonlinear characteristics of the process units with respect to the recycled impurity amounts (case study-3), the sequential ordering algorithm provides an interesting alternative for decomposing the overall NLP problem. The solutions may be by definition suboptimal compared to the overall NLP problem, but are significantly easier to compute, especially in practical applications where commercial process simulators are used, which are cumbersome to introduce into the NLP superstructure formulation. Although the potential is demonstrated in a simple case study, it is evident that further work is needed to optimally tune the use of the sequential ordering algorithm in this kind of problems of larger size. In this direction, one should try to minimize the optimality gap compared to the overall NLP problem and interpret the deviation from the solutions obtained by the linear problem representation.

The sequential ordering algorithm can be properly adapted and extended to cover more aspects of the process integration framework, including heat integration, property integration, and mass exchange networks. This will result in a unified framework for sequential process integration, which can be used in parallel with the superstructure problem formulations to enhance the realization and interpretability of the obtained solutions in complicated integration problems.

## References

- [1] Agrawal, R. and Sikdar, S.K. (2012) Energy, Environment and Sustainability Challenges and Opportunities for Chemical Engineers. *Current Opinion in Chemical Engineering*, **1**, 201-203. <http://dx.doi.org/10.1016/j.coche.2012.07.002>
- [2] Gwehenberger, G. and Narodoslawsky, M. (2008) Sustainable Processes—The Challenge of the 21st Century for Chemical Engineering. *Process Safety and Environmental Protection*, **86**, 321-327. <http://dx.doi.org/10.1016/j.psep.2008.03.004>
- [3] Klatt, K.-U. and Marquardt, W. (2009) Perspectives for Process Systems Engineering—Personal Views from Academia and Industry. *Computers & Chemical Engineering*, **33**, 536-550. <http://dx.doi.org/10.1016/j.compchemeng.2008.09.002>
- [4] Grossmann, I.E. and Guillen-Gosalbez, G. (2010) Scope for the Application of Mathematical Programming Techniques in the Synthesis and Planning of Sustainable Processes. *Computers & Chemical Engineering*, **34**, 1365-1376. <http://dx.doi.org/10.1016/j.compchemeng.2009.11.012>
- [5] Dunn, R.F. and El-Halwagi, M.M. (2003) Process Integration Technology Review: Background and Applications in the Chemical Process Industry. *Journal of Chemical Technology and Biotechnology*, **78**, 1011-1021.

- <http://dx.doi.org/10.1002/jctb.738>
- [6] El-Halwagi, M.M. (2006) Process Integration. Academic Press/Elsevier, San Diego.
- [7] El-Halwagi, M.M. and Manousiouthakis, V. (1990) Automatic Synthesis of Mass-Exchange Networks with Single-Component Targets. *Chemical Engineering Science*, **45**, 2813-2831. [http://dx.doi.org/10.1016/0009-2509\(90\)80175-E](http://dx.doi.org/10.1016/0009-2509(90)80175-E)
- [8] El-Halwagi, M.M., Gabriel, F. and Harell, D. (2003) Rigorous Graphical Targeting for Resource Conservation via Material Recycle/Reuse Networks. *Industrial & Engineering Chemistry Research*, **42**, 4319-4328. <http://dx.doi.org/10.1021/ie030318a>
- [9] Hallale, N. (2002) A New Graphical Targeting Method for Water Minimisation. *Advances in Environmental Research*, **6**, 377-390. [http://dx.doi.org/10.1016/S1093-0191\(01\)00116-2](http://dx.doi.org/10.1016/S1093-0191(01)00116-2)
- [10] Wang, Y.P. and Smith, R. (1994) Waste-Water Minimization (Vol 49, Pg 981, 1994). *Chemical Engineering Science*, **49**, 3533-3533.
- [11] Foo, D.C.Y., Manan, Z.A. and Tan, Y.L. (2006) Use Cascade Analysis to Optimize Water Networks. *Chemical Engineering Progress*, **102**, 45-52.
- [12] Manan, Z.A., Tan, Y.L. and Foo, D.C.Y. (2004) Targeting the Minimum Water Flow Rate Using Water Cascade Analysis Technique. *AIChE Journal*, **50**, 3169-3183. <http://dx.doi.org/10.1002/aic.10235>
- [13] Sorin, M. and Bedard, S. (1999) The Global Pinch Point in Water Reuse Networks. *Process Safety and Environmental Protection*, **77**, 305-308. <http://dx.doi.org/10.1205/095758299530189>
- [14] Adekola, O. and Majozi, T. (2011) Wastewater Minimization in Multipurpose Batch Plants with a Regeneration Unit: Multiple Contaminants. *Computers & Chemical Engineering*, **35**, 2824-2836. <http://dx.doi.org/10.1016/j.compchemeng.2011.04.008>
- [15] Gonws, J.F., Majozi, T., Foo, D.C.Y., Chen, C.L. and Lee, J.Y. (2010) Water Minimization Techniques for Batch Processes. *Industrial & Engineering Chemistry Research*, **49**, 8877-8893. <http://dx.doi.org/10.1021/ie100130a>
- [16] Majozi, T. (2005) An Effective Technique for Wastewater Minimisation in Batch Processes. *Journal of Cleaner Production*, **13**, 1374-1380. <http://dx.doi.org/10.1016/j.jclepro.2005.04.016>
- [17] Majozi, T., Brouckaert, C.J. and Buckley, C.A. (2006) A Graphical Technique for Wastewater Minimisation in Batch Processes. *Journal of Environmental Management*, **78**, 317-329. <http://dx.doi.org/10.1016/j.jenvman.2005.04.026>
- [18] Bagajewicz, M. and Savelski, M. (2001) On the Use of Linear Models for the Design of Water Utilization Systems in Process Plants with a Single Contaminant. *Chemical Engineering Research & Design*, **79**, 600-610. <http://dx.doi.org/10.1205/02638760152424389>
- [19] Gabriel, F.B. and El-Halwagi, M.M. (2005) Simultaneous Synthesis of Waste Interception and Material Reuse Networks: Problem Reformulation for Global Optimization. *Environmental Progress*, **24**, 171-180. <http://dx.doi.org/10.1002/ep.10081>
- [20] Alva-Argaez, A., Kokossis, A.C. and Smith, R. (2007) A Conceptual Decomposition of Minlp Models for the Design of Water-Using Systems. *International Journal of Environment and Pollution*, **29**, 177-205. <http://dx.doi.org/10.1504/IJEP.2007.012803>
- [21] Faria, D.C. and Bagajewicz, M.J. (2008) A New Approach for the Design of Multicomponent Water Wastewater Networks. *18th European Symposium on Computer Aided Process Engineering—ESCAPE 18*, Lyon, 1-4 June 2008.
- [22] Karuppiah, R. and Grossmann, I.E. (2006) Global Optimization for the Synthesis of Integrated Water Systems in Chemical Processes. *Computers & Chemical Engineering*, **30**, 650-673. <http://dx.doi.org/10.1016/j.compchemeng.2005.11.005>
- [23] Ahmetovic, E. and Grossmann, I.E. (2010) Strategies for the Global Optimization of Integrated Process Water Networks. In: Pierucci, S. and Ferraris, B.G., Eds., *20th European Symposium on Computer Aided Process Engineering*, Elsevier, Ischia, 901-906. [http://dx.doi.org/10.1016/S1570-7946\(10\)28151-8](http://dx.doi.org/10.1016/S1570-7946(10)28151-8)
- [24] Boix, M., Montastruc, L., Pibouleau, L., Azzaro-Pantel, C. and Domenech, S. (2011) A Multiobjective Optimization Framework for Multicontaminant Industrial Water Network Design. *Journal of Environmental Management*, **92**, 1802-1808. <http://dx.doi.org/10.1016/j.jenvman.2011.02.016>
- [25] Napoles-Rivera, F., Ponce-Ortega, J.M., El-Halwagi, M.M. and Jimenez-Gutierrez, A. (2012) Global Optimization of Wastewater Integration Networks for Processes with Multiple Contaminants. *Environmental Progress & Sustainable Energy*, **31**, 449-458. <http://dx.doi.org/10.1002/ep.10557>
- [26] Tudor, R. and Lavric, V. (2011) Dual-Objective Optimization of Integrated Water/Wastewater Networks. *Computers & Chemical Engineering*, **35**, 2853-2866. <http://dx.doi.org/10.1016/j.compchemeng.2011.04.010>
- [27] Kheireddine, H., Dadmohammadi, Y., Deng, C., Feng, X.A. and El-Halwagi, M. (2011) Optimization of Direct Re-

- cycle Networks with the Simultaneous Consideration of Property, Mass, and Thermal Effects. *Industrial & Engineering Chemistry Research*, **50**, 3754-3762. <http://dx.doi.org/10.1021/ie1012272>
- [28] Bishnu, S.K., Linke, P., Alnouri, S.Y. and El-Halwagi, M. (2014) Multiperiod Planning of Optimal Industrial City Direct Water Reuse Networks. *Industrial & Engineering Chemistry Research*, **53**, 8844-8865. <http://dx.doi.org/10.1021/ie5008932>
- [29] Baykasoglu, A. (2005) Preemptive Goal Programming Using Simulated Annealing. *Engineering Optimization*, **37**, 49-63. <http://dx.doi.org/10.1080/0305215042000268606>

## Supplement. Electronic Supplementary Information (ESI)

### S.1. Global Optimality of a Group of Ordered Process Sinks

As mentioned in the manuscript (Section 2.3), the sequential ordering algorithm does not guarantee global optimality for a given cardinality of ordered process sinks (*i.e.*, with the exception of the first and last step of the algorithm, namely when only the first sink or all sinks are ordered). In the manuscript, we stated one simple condition (Equations (9)) that can be used to identify groups of process sinks that may outperform, from a global optimality perspective, those selected by the sequential ordering algorithm for the same cardinality. We present the following two examples to illustrate the application of this condition.

#### Example-1

Let us consider the sets of process sinks and sources in **Table S1**. Applying the sequential ordering algorithm (*i.e.*, not presented here in detail), the optimal ranking at the second selection step is  $SK_{sel} = \{SK1, SK3\}$ . The respective recycled amount is 142.8 mol/sec and the amount of impurity is 32.1 mol/sec.

The sum of the required amounts of the target compound at the process sinks SK2 and SK3 exceeds the corresponding recycled amount at the ordered sinks SK1 and SK3 of the sequential algorithm, namely:

$$G(SK_2) + G(SK_3) = 185 > 142.8 = R_1 + R_{3/1}$$

Therefore, the couple of process sinks  $\{SK2, SK3\}$  is a potential group that outperforms the selected group of sinks by the ordering algorithm (*i.e.*,  $SK_{sel} = \{SK1, SK3\}$ ). Indeed, it can be easily verified, for instance, that:

$$R_2 + R_{3/2} = 145.6 > 142.8 = R_1 + R_{3/1}$$

#### Example-2

Let us consider the sets of process sinks and sources in **Table S2**. This set comprises the same set of process sinks and sources like in example-1, the sources having now different loads. Applying the sequential ordering algorithm (*i.e.*, not presented here in detail), the optimal ranking at the second selection step is  $SK_{sel} = \{SK1, SK2\}$ . The respective recycled amount is 190.4 mol/sec and the amount of impurity is 31.1 mol/sec.

The sum of the required amounts of the target compound at the process sinks SK3 and SK1 exceeds the corresponding recycled amount at the ordered sinks SK1 and SK3 of the sequential algorithm, namely:

$$G(SK_3) + G(SK_1) = 240 > 190.4 = R_1 + R_{2/1}$$

Therefore, the couple of sinks  $\{SK3, SK1\}$  is a potential group that outperforms the selected group of sinks by the ordering algorithm (*i.e.*,  $SK_{sel} = \{SK1, SK2\}$ ). In this case, it can be easily verified that the maximum amount that can be recycled to  $\{SK3, SK1\}$  is 190 mol/sec.

**Table S1.** Process sinks and sources of example-1.

	Process sinks		
	Required amount of target compound (mol/sec)	Maximum allowable impurity (% mol/mol)	Maximum allowable impurity (mol/sec)
SK1	150	10	15.0
SK2	95	17	16.2
SK3	90	19	17.1
SK4	80	2.5	2.0
	Process sources		
	Waste load mol/sec	Impurity content (% mol/mol)	Impurity content (mol/sec)
SR1	100	15	15
SR2	90	40	36
SR3	120	50	60

**Table S2.** Process sinks and sources of example-2.

	Process sinks		
	Required amount of target compound (mol/sec)	Maximum allowable impurity (% mol/mol)	Maximum allowable impurity (mol/sec)
SK1	150	10	15.0
SK2	95	17	16.2
SK3	90	19	17.1
SK4	80	2.5	2.0
	Process sources		
	Waste load mol/sec	Impurity content (% mol/mol)	Impurity content (mol/sec)
SR1	180	15	27
SR2	30	40	12
SR3	30	50	15

## S.2. Statistical Evaluation of All Categories of Scenarios for the Optimality Performance of the Sequential Ordering Algorithm

**Tables S3-S5** present the statistical evaluation of the indices *Ind1*, *Ind2*, and *Ind3*, respectively, according to their definition in the manuscript for each category of scenarios. Diverse categories of scenarios were identified based on factors and their levels of differentiation described in **Table 2** of the manuscript. The number of scenarios for each individual category refers to combinations of all the factors not characterizing the specific scenario.

For instance, the number of scenarios for the P2\_10 problems refers to the case of a 10% remaining (*i.e.*, 90% depleting) target compound in the process sources, the P2\_30 problems refers to the case of a 30% remaining (*i.e.*, 70% depleting) target compound in the process sources, and so on. Thus, summing up the number of scenarios for the problems for P2\_10 up to P2\_90 (*i.e.*, 6900 for each type of these problems) results in the total number of scenarios for the P2 (all problems) case, namely 34500 scenarios.

**Table S3.** Statistical evaluation of *Ind1* according to the mean, median, standard deviation over the whole scenarios interval, as well as the trimmed mean over the 5% - 95% percentiles interval.

Categories of scenarios	<i>Ind1</i>			
	Mean	Median	Standard deviation	Trimmed mean over [5% - 95%] percentiles
All problems (75,900 scenarios)	69	2	244	22
P1 (all problems, 6,900 scenarios)	47	2	176	16
P2 (all problems, 34,500 scenarios)	113	4	319	53
P3 (all problems, 34,500 scenarios)	28	1	139	7
Sinks > sources (33,000 scenarios)	119	9	320	59
Sinks < sources (42,900 scenarios)	30	1	153	5
LH problems (25,300 scenarios)	60	2	229	17
Rand problems (25,300 scenarios)	70	2	248	23
Totrand problems (25,300 scenarios)	76	3	253	28
P2_10 problems (6900 scenarios)	179	9	433	103
P2_30 problems (6900 scenarios)	154	7	385	85
P2_50 problems (6900 scenarios)	111	5	288	58

**Continued**

P2_70 problems (6900 scenarios)	72	4	211	33
P2_90 problems (6900 scenarios)	51	2	181	19
P3_10 problems (6900 scenarios)	20	1	125	4
P3_30 problems (6900 scenarios)	22	1	122	4
P3_50 problems (6900 scenarios)	28	1	146	6
P3_70 problems (6900 scenarios)	32	1	147	8
P3_90 problems (6900 scenarios)	39	2	154	13
P1, sinks > sources (3000 scenarios)	86	8	249	39
P1, sinks < sources (3900 scenarios)	17	1	72	5
P2, sinks > sources (3000 scenarios)	189	24	403	119
P2, sinks < sources (3900 scenarios)	56	2	217	15
P3, sinks > sources (3000 scenarios)	56	4	202	20
P3, sinks < sources (3900 scenarios)	7	1	43	2
P1, LH problems (2300 scenarios)	33	2	120	13
P1, rand problems (2300 scenarios)	47	2	180	15
P1, totrand problems (2300 scenarios)	61	3	215	21
P2, LH problems (2300 scenarios)	111	4	316	51
P2, rand problems (2300 scenarios)	113	4	320	52
P2, totrand problems (2300 scenarios)	116	5	320	55
P3, LH problems (2300 scenarios)	14	1	90	3
P3, rand problems (2300 scenarios)	32	1	153	8
P3, totrand problems (2300 scenarios)	39	1	163	11
P2_10, sinks > sources (3000 scenarios)	297	54	533	212
P2_10, sinks < sources (3900 scenarios)	88	2	308	28
P2_30, sinks > sources (3000 scenarios)	250	30	480	170
P2_30, sinks < sources (3900 scenarios)	81	2	269	28
P2_50, sinks > sources (3000 scenarios)	190	28	366	130
P2_50, sinks < sources (3900 scenarios)	51	2	189	15
P2_70, sinks > sources (3000 scenarios)	120	15	274	72
P2_70, sinks < sources (3900 scenarios)	35	1	135	10
P2_90, sinks > sources (3000 scenarios)	86	9	243	42
P2_90, sinks < sources (3900 scenarios)	23	1	103	6
P3_10, sinks > sources (3000 scenarios)	43	2	186	11
P3_10, sinks < sources (3900 scenarios)	3	0	20	1
P3_30, sinks > sources (3000 scenarios)	45	3	178	14
P3_30, sinks < sources (3900 scenarios)	4	1	33	1
P3_50, sinks > sources (3000 scenarios)	57	4	214	18
P3_50, sinks < sources (3900 scenarios)	6	1	34	1
P3_70, sinks > sources (3000 scenarios)	64	4	212	25
P3_70, sinks < sources (3900 scenarios)	8	1	48	2
P3_90, sinks > sources (3000 scenarios)	74	7	215	34
P3_90, sinks < sources (3900 scenarios)	13	1	66	4

**Table S4.** Statistical evaluation of *Ind2* according to the mean, median, standard deviation over the whole scenarios interval, as well as the trimmed mean over the 5% - 95% percentiles interval.

Categories of scenarios	<i>Ind2</i>			
	Mean	Median	Standard deviation	Trimmed mean over [5% - 95%] percentiles
All problems (75,900 scenarios)	0.037	0.000	0.489	0.002
P1 (all problems, 6,900 scenarios)	0.049	0.000	0.391	0.006
P2 (all problems, 34,500 scenarios)	0.056	0.000	0.670	0.006
P3 (all problems, 34,500 scenarios)	0.016	0.000	0.212	0.000
Sinks > sources (33,000 scenarios)	0.063	0.000	0.718	0.006
Sinks < sources (42,900 scenarios)	0.018	0.000	0.160	0.000
LH problems (25,300 scenarios)	0.034	0.000	0.249	0.002
Rand problems (25,300 scenarios)	0.055	0.000	0.711	0.005
Totrand problems (25,300 scenarios)	0.023	0.000	0.386	0.000
P2_10 problems (6900 scenarios)	0.007	0.000	0.073	0.000
P2_30 problems (6900 scenarios)	0.049	0.000	0.597	0.004
P2_50 problems (6900 scenarios)	0.096	0.000	1.118	0.011
P2_70 problems (6900 scenarios)	0.072	0.000	0.646	0.012
P2_90 problems (6900 scenarios)	0.057	0.000	0.463	0.009
P3_10 problems (6900 scenarios)	0.001	0.000	0.020	0.000
P3_30 problems (6900 scenarios)	0.008	0.000	0.091	0.000
P3_50 problems (6900 scenarios)	0.011	0.000	0.087	0.000
P3_70 problems (6900 scenarios)	0.025	0.000	0.371	0.001
P3_90 problems (6900 scenarios)	0.037	0.000	0.265	0.004
P1, sinks > sources (3000 scenarios)	0.079	0.000	0.552	0.013
P1, sinks < sources (3900 scenarios)	0.026	0.000	0.187	0.002
P2, sinks > sources (3000 scenarios)	0.097	0.000	0.990	0.013
P2, sinks < sources (3900 scenarios)	0.025	0.000	0.197	0.001
P3, sinks > sources (3000 scenarios)	0.025	0.000	0.300	0.001
P3, sinks < sources (3900 scenarios)	0.009	0.000	0.101	0.000
P1, LH problems (2300 scenarios)	0.048	0.000	0.246	0.008
P1, rand problems (2300 scenarios)	0.060	0.000	0.355	0.010
P1, totrand problems (2300 scenarios)	0.039	0.000	0.522	0.001
P2, LH problems (2300 scenarios)	0.050	0.000	0.327	0.007
P2, rand problems (2300 scenarios)	0.084	0.000	0.991	0.009
P2, totrand problems (2300 scenarios)	0.035	0.000	0.508	0.002
P3, LH problems (2300 scenarios)	0.015	0.000	0.131	0.000
P3, rand problems (2300 scenarios)	0.026	0.000	0.322	0.002
P3, totrand problems (2300 scenarios)	0.008	0.000	0.119	0.000

**Continued**

P2_10, sinks > sources (3000 scenarios)	0.009	0.000	0.095	0.000
P2_10, sinks < sources (3900 scenarios)	0.006	0.000	0.049	0.000
P2_30, sinks > sources (3000 scenarios)	0.087	0.000	0.888	0.009
P2_30, sinks < sources (3900 scenarios)	0.020	0.000	0.143	0.001
P2_50, sinks > sources (3000 scenarios)	0.177	0.000	1.664	0.030
P2_50, sinks < sources (3900 scenarios)	0.034	0.000	0.268	0.003
P2_70, sinks > sources (3000 scenarios)	0.124	0.000	0.942	0.030
P2_70, sinks < sources (3900 scenarios)	0.032	0.000	0.229	0.003
P2_90, sinks > sources (3000 scenarios)	0.090	0.000	0.657	0.019
P2_90, sinks < sources (3900 scenarios)	0.032	0.000	0.214	0.003
P3_10, sinks > sources (3000 scenarios)	0.000	0.000	0.004	0.000
P3_10, sinks < sources (3900 scenarios)	0.001	0.000	0.026	0.000
P3_30, sinks > sources (3000 scenarios)	0.013	0.000	0.131	0.000
P3_30, sinks < sources (3900 scenarios)	0.004	0.000	0.035	0.000
P3_50, sinks > sources (3000 scenarios)	0.016	0.000	0.113	0.000
P3_50, sinks < sources (3900 scenarios)	0.007	0.000	0.060	0.000
P3_70, sinks > sources (3000 scenarios)	0.040	0.000	0.552	0.004
P3_70, sinks < sources (3900 scenarios)	0.012	0.000	0.095	0.000
P3_90, sinks > sources (3000 scenarios)	0.057	0.000	0.337	0.010
P3_90, sinks < sources (3900 scenarios)	0.022	0.000	0.190	0.001

**Table S5.** Statistical evaluation of *Ind3* according to the mean, median, standard deviation over the whole scenarios interval, as well as the trimmed mean over the 5% - 95% percentiles interval.

Categories of scenarios	<i>Ind3</i>			
	Mean	Median	Standard deviation	Trimmed mean over [5% - 95%] percentiles
All problems (75,900 scenarios)	0.002	0.000	0.019	0.0000
P1 (all problems, 6,900 scenarios)	0.003	0.000	0.025	0.0001
P2 (all problems, 34,500 scenarios)	0.002	0.000	0.016	0.0000
P3 (all problems, 34,500 scenarios)	0.002	0.000	0.021	0.0000
Sinks > sources (33,000 scenarios)	0.002	0.000	0.019	0.0001
Sinks < sources (42,900 scenarios)	0.002	0.000	0.020	0.0000
LH problems (25,300 scenarios)	0.001	0.000	0.011	0.0000
Rand problems (25,300 scenarios)	0.004	0.000	0.030	0.0001
Totrand problems (25,300 scenarios)	0.001	0.000	0.011	0.0000
P2_10 problems (6900 scenarios)	0.000	0.000	0.006	0.0000
P2_30 problems (6900 scenarios)	0.001	0.000	0.014	0.0000
P2_50 problems (6900 scenarios)	0.002	0.000	0.018	0.0001
P2_70 problems (6900 scenarios)	0.002	0.000	0.014	0.0002

## Continued

P2_90 problems (6900 scenarios)	0.003	0.000	0.022	0.0002
P3_10 problems (6900 scenarios)	0.000	0.000	0.011	0.0000
P3_30 problems (6900 scenarios)	0.002	0.000	0.022	0.0000
P3_50 problems (6900 scenarios)	0.002	0.000	0.023	0.0000
P3_70 problems (6900 scenarios)	0.003	0.000	0.024	0.0000
P3_90 problems (6900 scenarios)	0.003	0.000	0.024	0.0001
P1, sinks > sources (3000 scenarios)	0.004	0.000	0.025	0.0003
P1, sinks < sources (3900 scenarios)	0.003	0.000	0.024	0.0000
P2, sinks > sources (3000 scenarios)	0.002	0.000	0.014	0.0001
P2, sinks < sources (3900 scenarios)	0.002	0.000	0.017	0.0000
P3, sinks > sources (3000 scenarios)	0.002	0.000	0.022	0.0000
P3, sinks < sources (3900 scenarios)	0.002	0.000	0.021	0.0000
P1, LH problems (2300 scenarios)	0.002	0.000	0.016	0.0001
P1, rand problems (2300 scenarios)	0.006	0.000	0.037	0.0004
P1, totrand problems (2300 scenarios)	0.002	0.000	0.012	0.0000
P2, LH problems (2300 scenarios)	0.001	0.000	0.010	0.0000
P2, rand problems (2300 scenarios)	0.003	0.000	0.022	0.0001
P2, totrand problems (2300 scenarios)	0.001	0.000	0.013	0.0000
P3, LH problems (2300 scenarios)	0.001	0.000	0.011	0.0000
P3, rand problems (2300 scenarios)	0.005	0.000	0.034	0.0000
P3, totrand problems (2300 scenarios)	0.001	0.000	0.008	0.0000
P2_10, sinks > sources (3000 scenarios)	0.000	0.000	0.004	0.0000
P2_10, sinks < sources (3900 scenarios)	0.000	0.000	0.007	0.0000
P2_30, sinks > sources (3000 scenarios)	0.001	0.000	0.008	0.0000
P2_30, sinks < sources (3900 scenarios)	0.001	0.000	0.017	0.0000
P2_50, sinks > sources (3000 scenarios)	0.002	0.000	0.012	0.0003
P2_50, sinks < sources (3900 scenarios)	0.002	0.000	0.022	0.0000
P2_70, sinks > sources (3000 scenarios)	0.002	0.000	0.013	0.0004
P2_70, sinks < sources (3900 scenarios)	0.002	0.000	0.015	0.0000
P2_90, sinks > sources (3000 scenarios)	0.004	0.000	0.025	0.0004
P2_90, sinks < sources (3900 scenarios)	0.002	0.000	0.020	0.0001
P3_10, sinks > sources (3000 scenarios)	0.000	0.000	0.007	0.0000
P3_10, sinks < sources (3900 scenarios)	0.001	0.000	0.013	0.0000
P3_30, sinks > sources (3000 scenarios)	0.001	0.000	0.018	0.0000
P3_30, sinks < sources (3900 scenarios)	0.002	0.000	0.025	0.0000
P3_50, sinks > sources (3000 scenarios)	0.003	0.000	0.027	0.0000
P3_50, sinks < sources (3900 scenarios)	0.002	0.000	0.019	0.0000
P3_70, sinks > sources (3000 scenarios)	0.4	0.000	0.027	0.0001
P3_70, sinks < sources (3900 scenarios)	0.003	0.000	0.022	0.0000
P3_90, sinks > sources (3000 scenarios)	0.003	0.000	0.024	0.0003
P3_90, sinks < sources (3900 scenarios)	0.003	0.000	0.024	0.0000

### S.3. Stream Table of Case Study-3

The stream flows and temperatures for the case study-3 of the manuscript are presented in **Table S6**. The data refer to a base case before process integration through direct recycling. Full model information (e.g., molecular structures of compounds A, B, C, and D, reaction kinetic model and NRTL parameters) can be provided by the authors upon request.

**Table S6.** Flows and temperatures of the streams presented in **Figure 8** of the manuscript for Case Study-3. A, B, C and D refer to compounds participating in the reaction and Sol1 and Sol2 refer to the two solvents used in the process. Empty cell designate the absence of the compounds in the respective streams.

Streams	Temperature (°C)	A kmol/hr	B kmol/hr	C kmol/hr	D kmol/hr	Sol1 kmol/hr	Sol2 kmol/hr
1	20	13.0					
2	20	15.0	10.0		10.0		
3	20	10.0	5.0		10.0	30.0	10.0
4	20	10.0			20.0	10.0	10.0
5	20		10.0				
6	20	13.0	10.0				
7	20	20.5	15.0		5.0		
8	150	17.3	11.8	3.2	11.3		
9	132	19.3	13.0	3.2	16.1	1.0	1.0
10	150	0.1	0.0	0.0	7.0	0.9	0.6
11	150	19.3	13.0	3.2	9.1	0.2	0.4
12	20	7.5	5.0				5.0
13	20	7.5	5.0				5.0
14	20	22.5	12.5		25.0	35.0	15.0
15	50	2.0	1.1		4.8	1.0	1.0
16	50	20.5	11.4		20.2	34.0	14.0
17	20	5.0	2.5		10.0	5.0	5.0
18	20	5.0	2.5		10.0	5.0	5.0
19	70	0.0	0.0		1.0	3.8	1.1
20	70	5.0	2.5		9.0	1.2	3.9