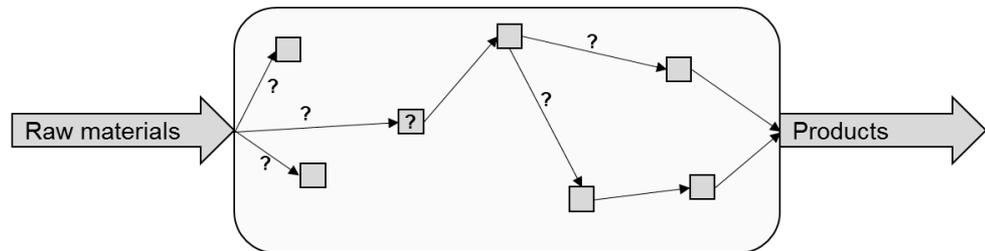
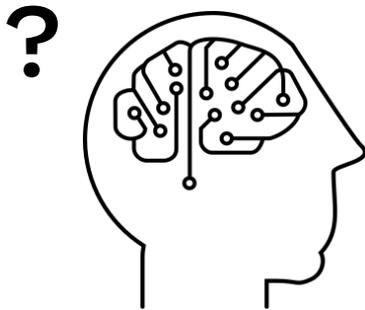


SYNTHÈSE AUTOMATIQUE DE PROCÉDÉS PAR APPROCHE ÉVOLUTIONNAIRE

Thibaut Neveux



Journées Promotion Procédés Produits (J3P) – 09/11/2017 – Nancy

SOMMAIRE



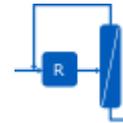
Introduction

Approches pour la
synthèse de
procédés



Approche évolutionnaire

Construction automatique
de flowsheets



Etude de cas

Réaction-séparation-
recyclage



Conclusion

Perspectives pour
ces approches

SOMMAIRE

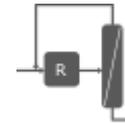


Introduction

Approches pour la
synthèse de
procédés



Approche évolutionnaire



Etude de cas

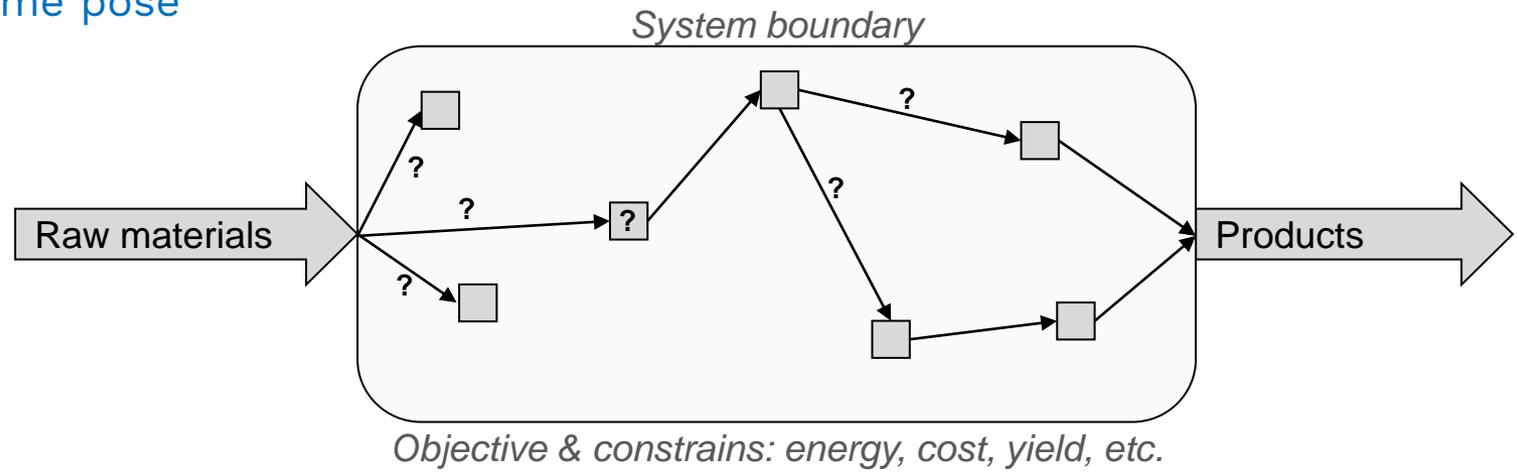


Conclusion

SYNTHÈSE DE PROCÉDÉS

“Process synthesis is very much the fun part of engineering. It is where one invents the structure and operating levels for a new chemical manufacturing process.” A.W. Westerberg

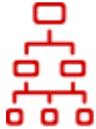
Problème posé



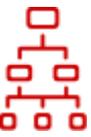
Synthèse de procédé = détermination

- ✓ De la structure du procédé (flowsheet)
- ✓ Des degrés de liberté associés (e.g. dimensions, conditions opératoires)

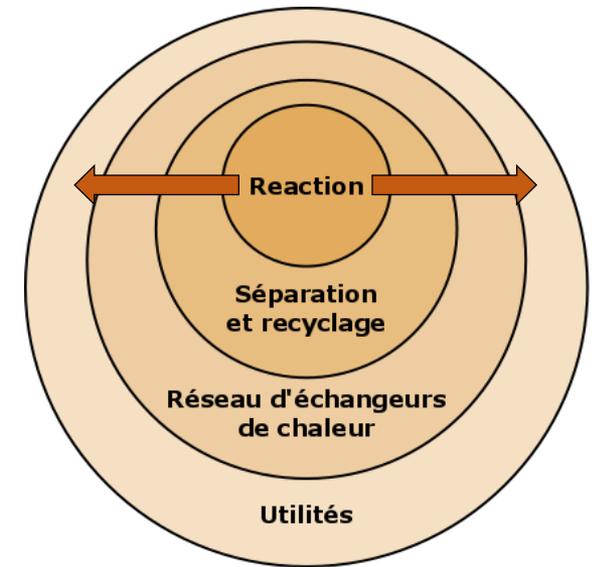
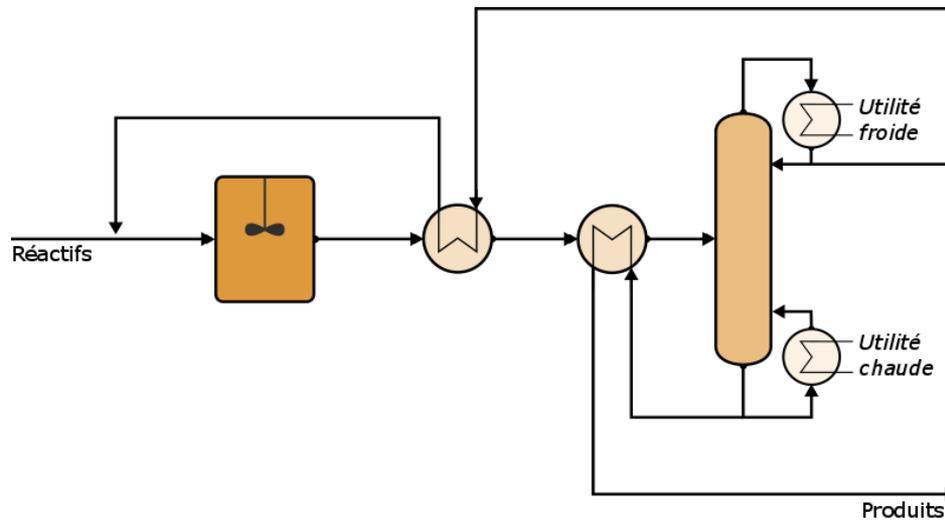
Classification des approches

-  Décomposition hiérarchique
-  Optimisation numérique

Westerberg (2004) ; Cremashi (2015)



- Décomposition du problème et résolution séquentielle
- Hiérarchie décisionnelle (Douglas,1988), *Onion-model* (Smith & Linnhoff, 1988), etc.



- Approche efficace
- Solutions acceptables obtenues "rapidement"

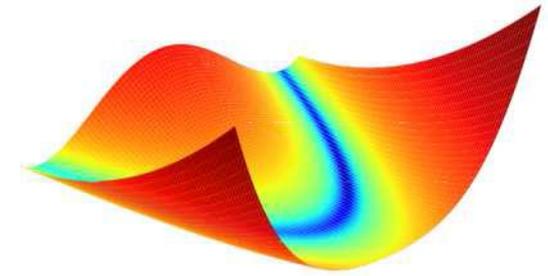


Pas d'interactions entre niveaux
→ Solutions potentiellement éloignées de l'optimum



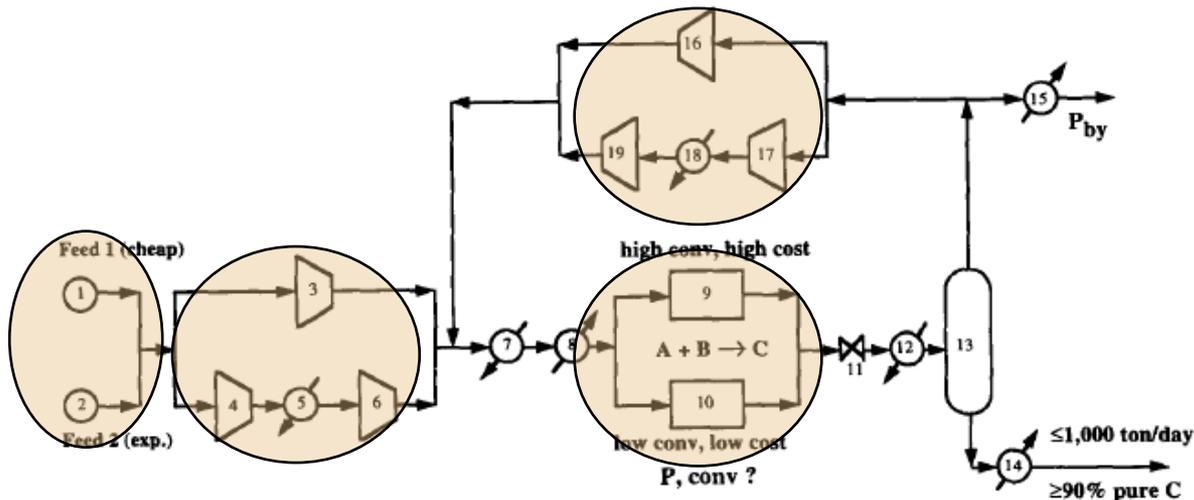
A structure figée

- Postulat d'un schéma de procédé (expertise, heuristiques, décomposition hiérarchique)
- Degrés de libertés obtenus par optimisation (SQP, point-intérieur, etc.)



Optimisation structurelle

- Définition d'un ensemble d'alternatives → représentation par une **superstructure**
- Optimisation de la structure et des variables continues (GDP, MINLP)



La solution optimale doit se situer dans l'ensemble prédéfini

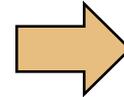
CHALLENGES

Complexité croissante

- des opérations unitaires et leurs degrés de liberté → intensification locale
- des structures de procédés (intégrations avancées) → intensification globale
- des performances attendues (énergie, environnement, coûts, société etc.)

Encore beaucoup d'expertise et d'intuition dans les méthodes de synthèse

Difficile d'aborder toute la complexité de la synthèse et proposer des structures de procédé



Objectif de l'étude :
s'affranchir d'une partie de cette complexité



Possibilité d'un algorithme proposant une architecture optimale *ab-initio* ?
C'est-à-dire sans présupposer d'alternatives (superstructure)

Idéalement on fournirait

- Une **bibliothèque d'opérations unitaires à disposition**
- Une formulation réaliste prenant en compte les différents objectifs et contraintes
- Des limites au système : flux d'entrées (matière, énergie), utilités disponibles, etc.

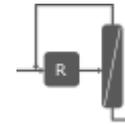
SOMMAIRE



Introduction



Approche évolutionnaire
Construction automatique
de flowsheets



Etude de cas



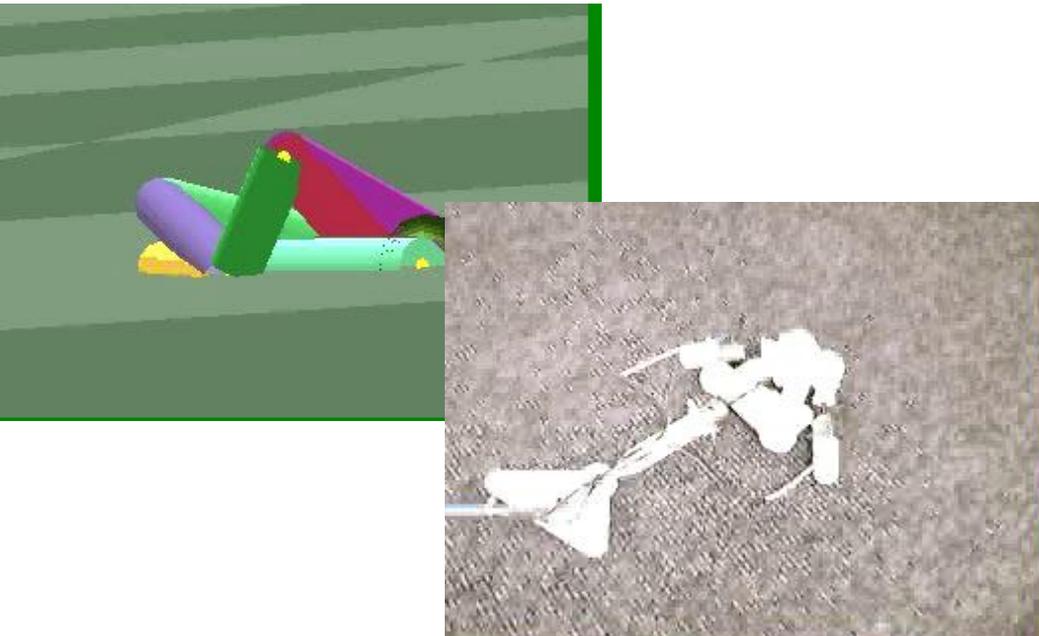
Conclusion

VOIE INVESTIGUÉE : PROGRAMMATION ÉVOLUTIONNAIRE

Approche : transposer un projet de robotique au génie des procédés

Projet GOLEM [8]

- Objectif : concevoir un robot qui avance dans le sable
- Moteur physique couplé à un calcul évolutionnaire



Représentation du robot
Modules de différents type,
connectés par des **jonctions**



Représentation du procédé
Opérations unitaires,
connectés par des **courants**

Moteur physique
(calcul de la motricité)



Simulateur / optimisateur
de procédés
(calcul de la performance)

VUE GÉNÉRALE

Classes d'algorithmes évolutionnaires

- Algorithme génétique (GA)
- **Programmation Évolutionnaire (EP)**
- Programmation Génétique (GP)
- Stratégies d'Evolution (ES)

○ Population initiale

Par exemple un "tuyau"



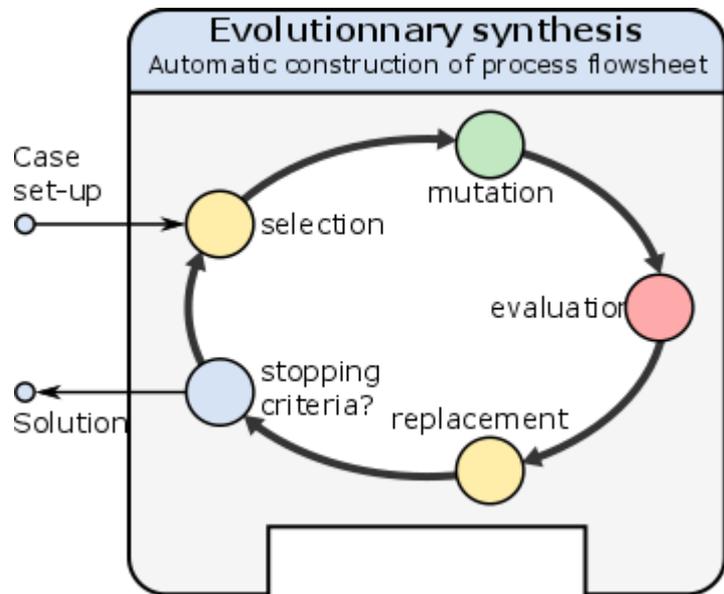
- Sélection de certains individus (procédés)
(sélection aléatoire, basée sur le rang, sur la valeur du critère, etc.)

- Application d'une ou plusieurs mutations
(cf. après)

- Evaluation des mutants
Calcul de leur(s) fonction(s) objectif(s)

- Remplacement de certains individus par ces mutants

- Convergence ?





OPÉRATIONS ÉLÉMENTAIRES DE MUTATION

Programmation évolutionnaire : mutation au cœur du processus

Définition d'opérateurs de mutation

- Aussi élémentaires que possible
- Tout procédé doit pouvoir se construire par application successive des opérateurs

3 opérateurs

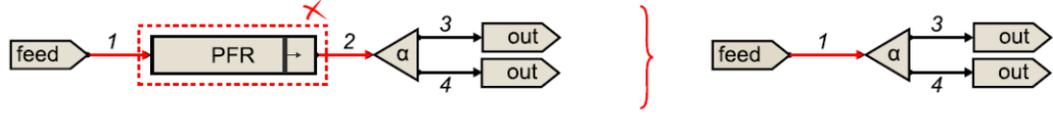
- Probabilité d'occurrence prédéfinies
- Par exemple :
 - 5% pour ajout d'une opération
 - 5% pour suppression d'une opération
 - 20% pour permutation de courants

Mutation operator: block addition



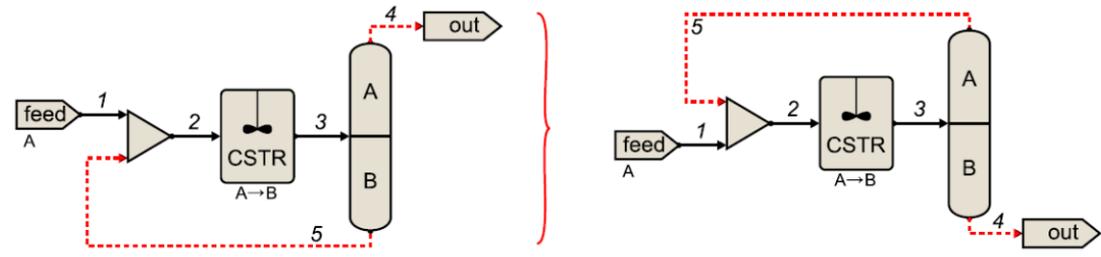
- a) randomly select a process stream (here stream 1)
- b) randomly select a type of block (here a mixer)
- c) add the block and a new connecting stream

Mutation operator: block removal



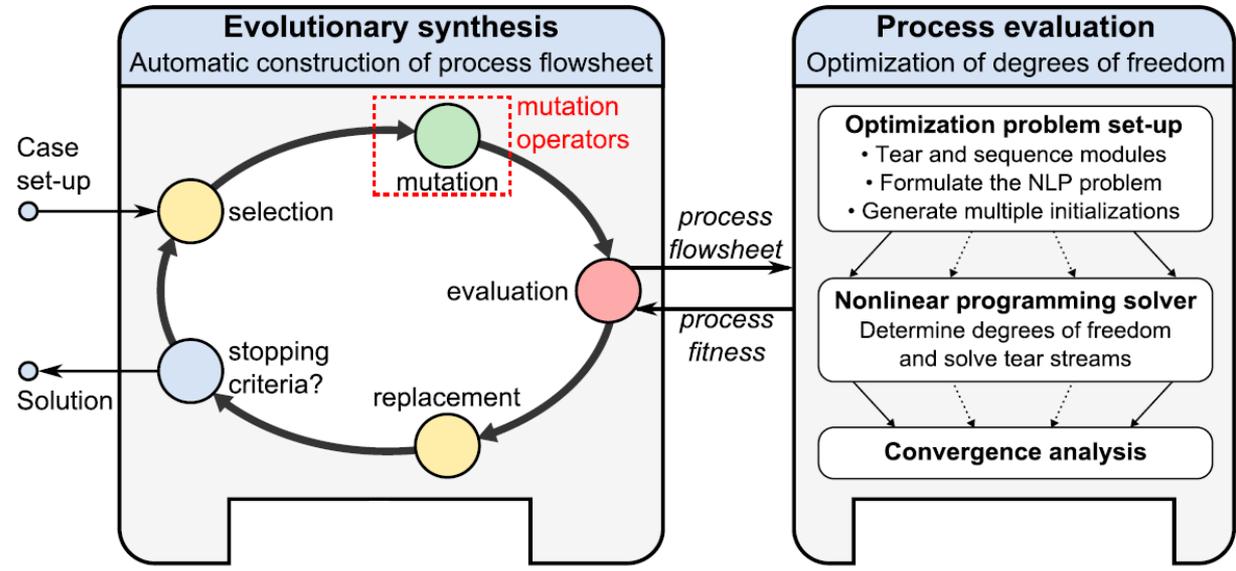
- a) randomly select a process block
- b) remove it and merge the corresponding streams

Mutation operator: permutation of two streams

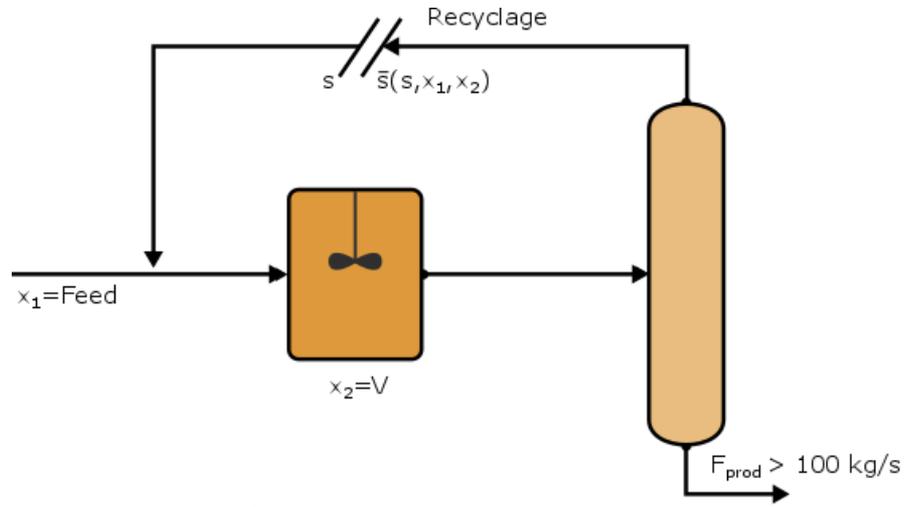


- a) randomly select two process streams (here 4 and 5)
- b) permute their destinations or their origins

EVALUATION DU SCHÉMA DE PROCÉDÉ

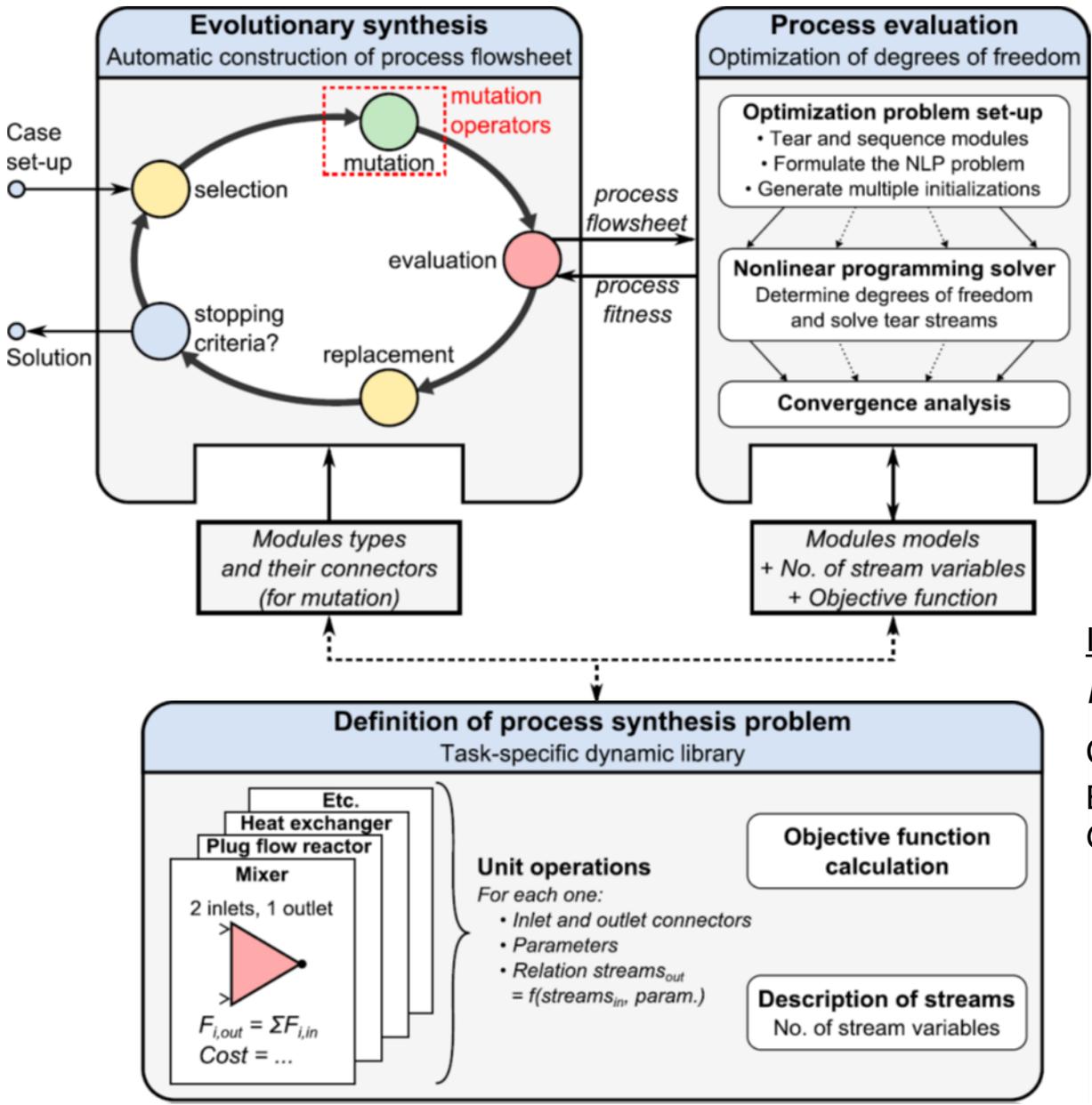


$$\left\{ \begin{array}{l}
 \max \quad F_{obj}(\mathbf{z}) \\
 \text{s.t.} \quad \mathbf{g}(\mathbf{z}) \leq \mathbf{0} \\
 \quad \quad \mathbf{h}(\mathbf{z}) = \mathbf{0} \\
 \quad \quad \mathbf{t}(\mathbf{z}) = \mathbf{s} - \bar{\mathbf{s}}(\mathbf{z}) = \mathbf{0} \\
 \quad \quad \mathbf{z} \in [\mathbf{z}_l, \mathbf{z}_u] \\
 \quad \quad \mathbf{z}^T = [\mathbf{x}^T, \mathbf{s}^T]
 \end{array} \right.$$



Résolution par un solveur SQP (SLSQP, NLPQLP, etc.)
Unfeasible path (gestion des recyclage dans le problème d'optimisation)

DÉFINITION D'UN PROBLÈME



Implémentation de la méthode
PSEvo (Process Synthesis by Evolution)

Codé en Fortran 2008

Bibliothèques externes : Graphviz,
 Gnuplot, SLSQP (ou autre solveur)

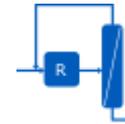
SOMMAIRE



Introduction



Approche évolutionnaire



Etude de cas
Réaction-séparation-
recyclage



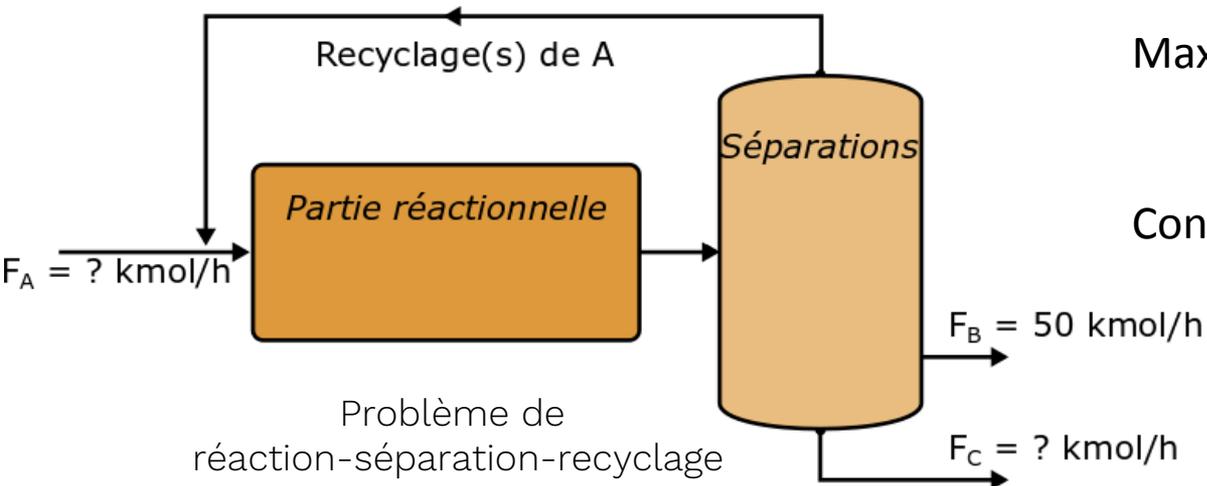
Conclusion

PRÉSENTATION DE L'EXEMPLE

Etude de cas : comparaison aux résultats de Kokossis et Floudas (1991)
→ Etude de la production de monochlorobenzène par chloration du benzène

Simplification des auteurs (Kokossis et Floudas, 1991)

- Schéma réactionnel $A \xrightarrow{k_1} B \xrightarrow{k_2} C$
- Simplifications structurelles
 - Injection de Cl_2 et stripping HCl en amont/aval de chaque réacteur
 - Même température en entrées/sorties de chaque unité → échangeurs intégrés dans chaque opération
- Réacteurs : CSTR et PFR
- Séparateurs : distillation avec pureté 100% (4 cas possibles A/BC, AB/C, A/B, B/C)

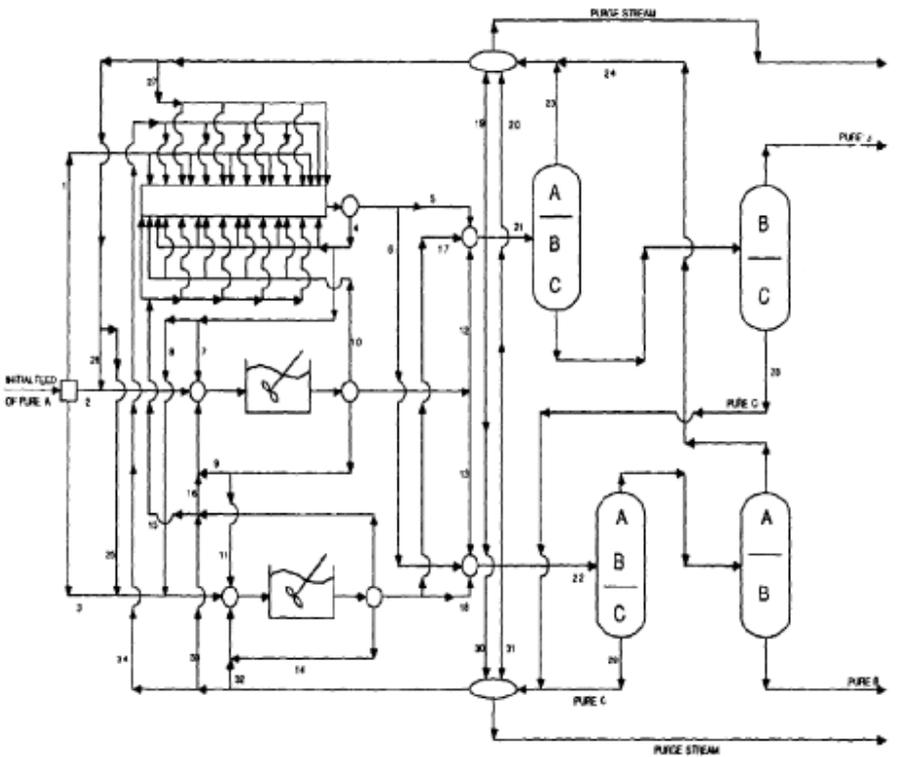


Max Profit = vente_B - achat_A
- CAPEX_{actualisé}
- OPEX_{utilités}
Contrainte : production 50 mol/h B pur

DÉFINITION DU PROBLÈME

Superstructure-based process synthesis

Definition of a superstructure
(here from Kokossis & Floudas)

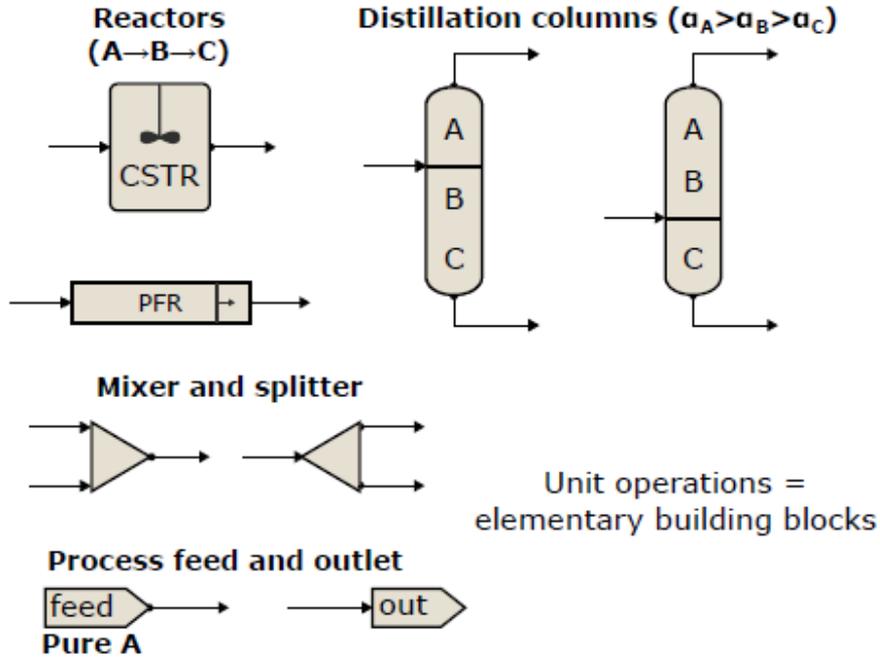


Solved by Mixed-Integer Non-Linear Programming (MINLP)

Selection of the optimal structure among the predefined alternatives

Ab-initio process synthesis (this work)

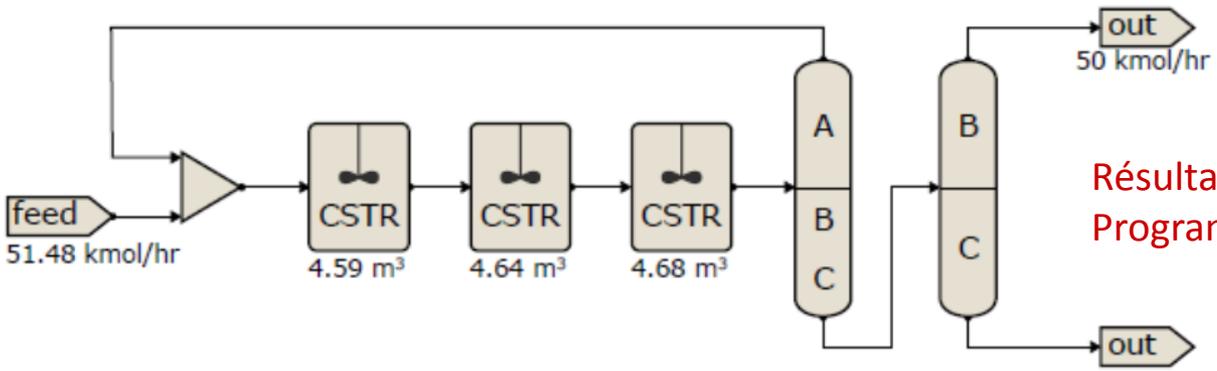
Definition of available unit operations,
no superstructure assumed



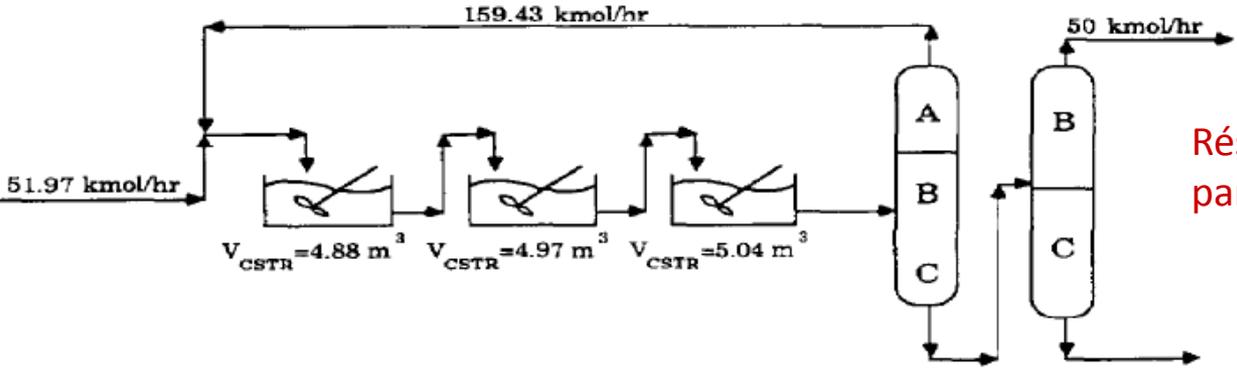
Solved by Evolutionary Programming (EP) and Non-Linear Programming (NLP)

Optimal structure generated by choosing, assembling and connecting unit operations (search space is virtually infinite)

RÉSULTATS TROUVÉS

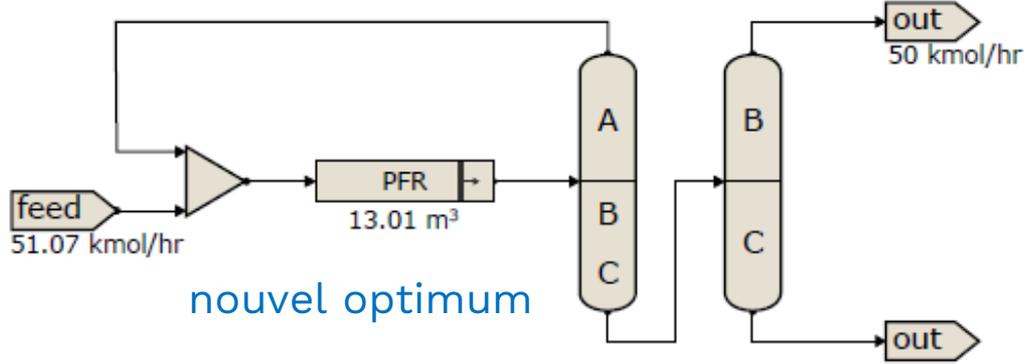


Résultat obtenu par Programmation évolutionnaire



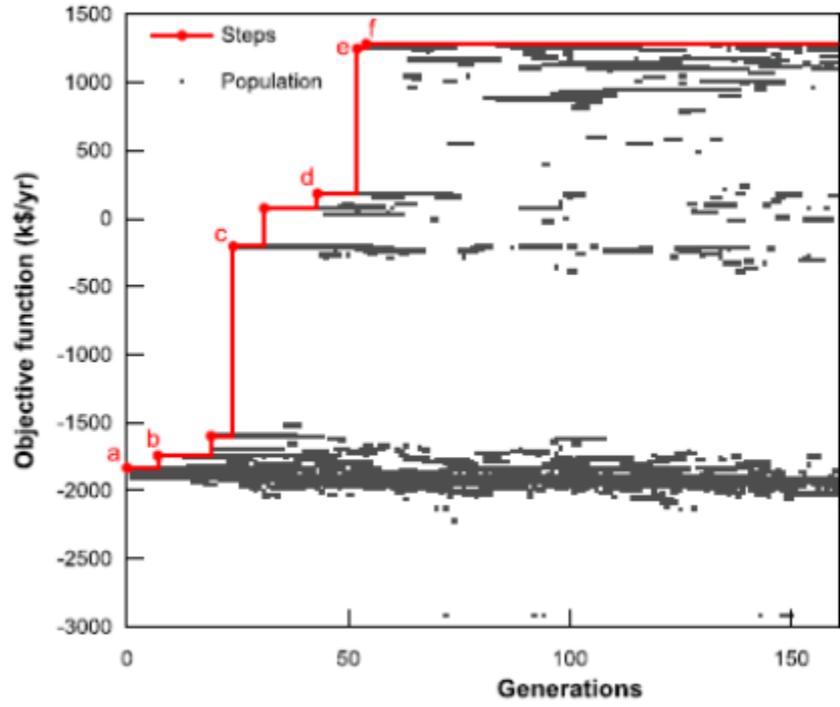
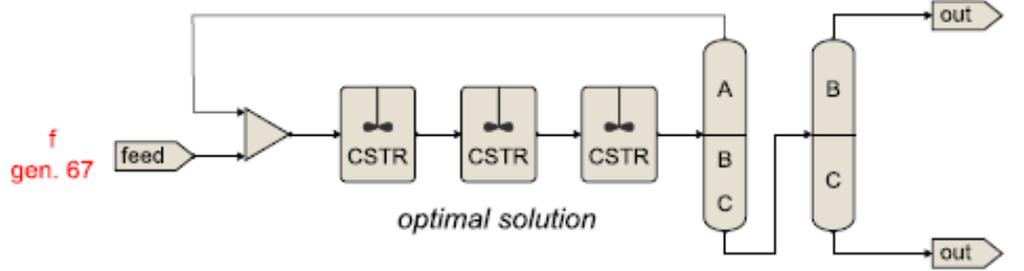
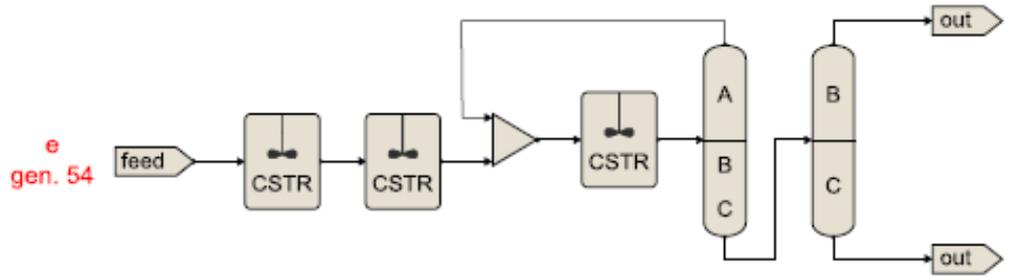
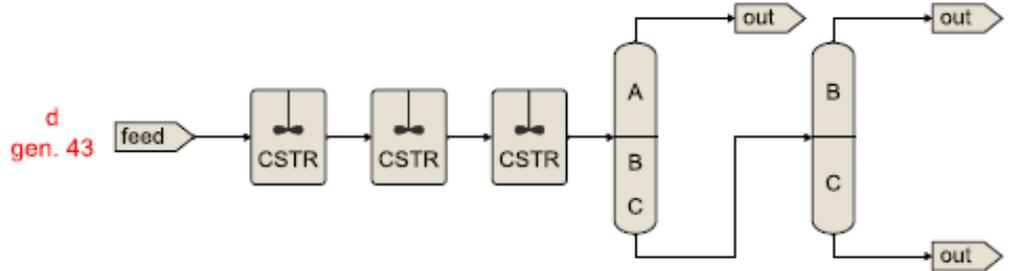
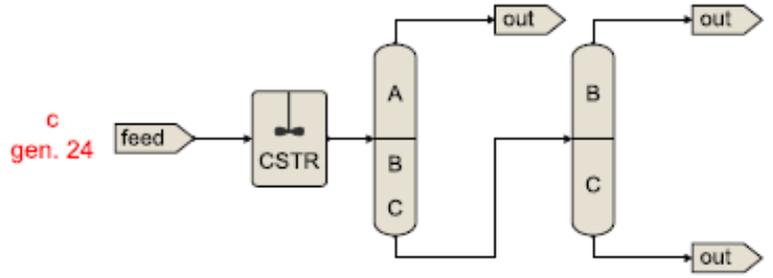
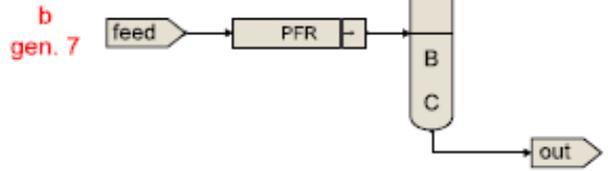
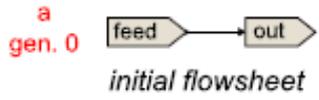
Résultat de Kokossis & Floudas (1991) par optimisation d'une superstructure

N.B. si $CAPEX_{CSTR} = CAPEX_{PFR}$ →

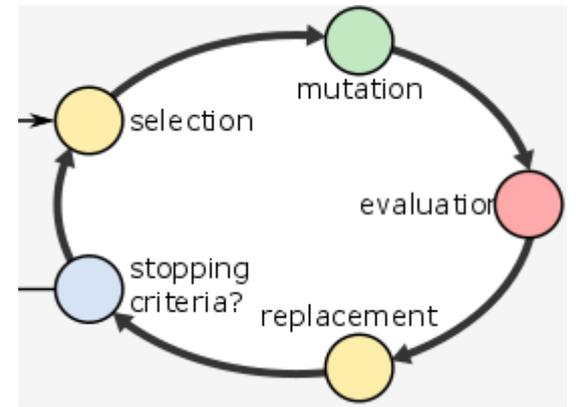
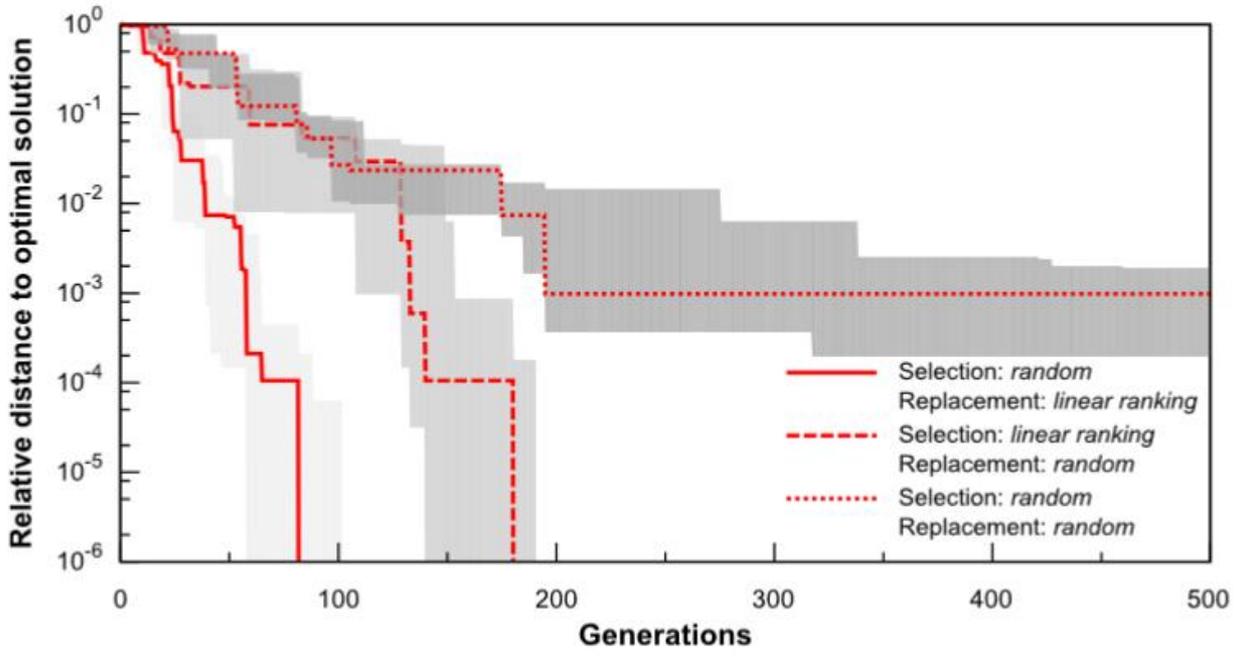
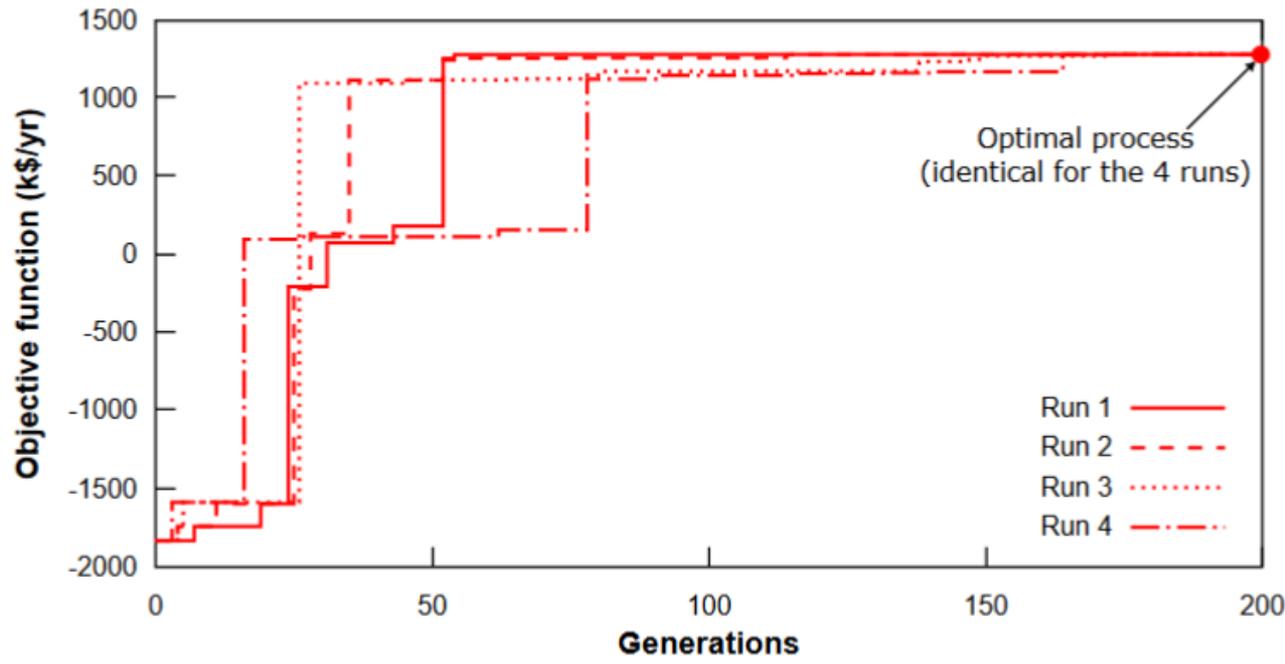


nouvel optimum

EXEMPLE D'ÉVOLUTION



ANALYSE DE SENSIBILITÉ



CONVERGENCE ET (BIO)DIVERSITÉ

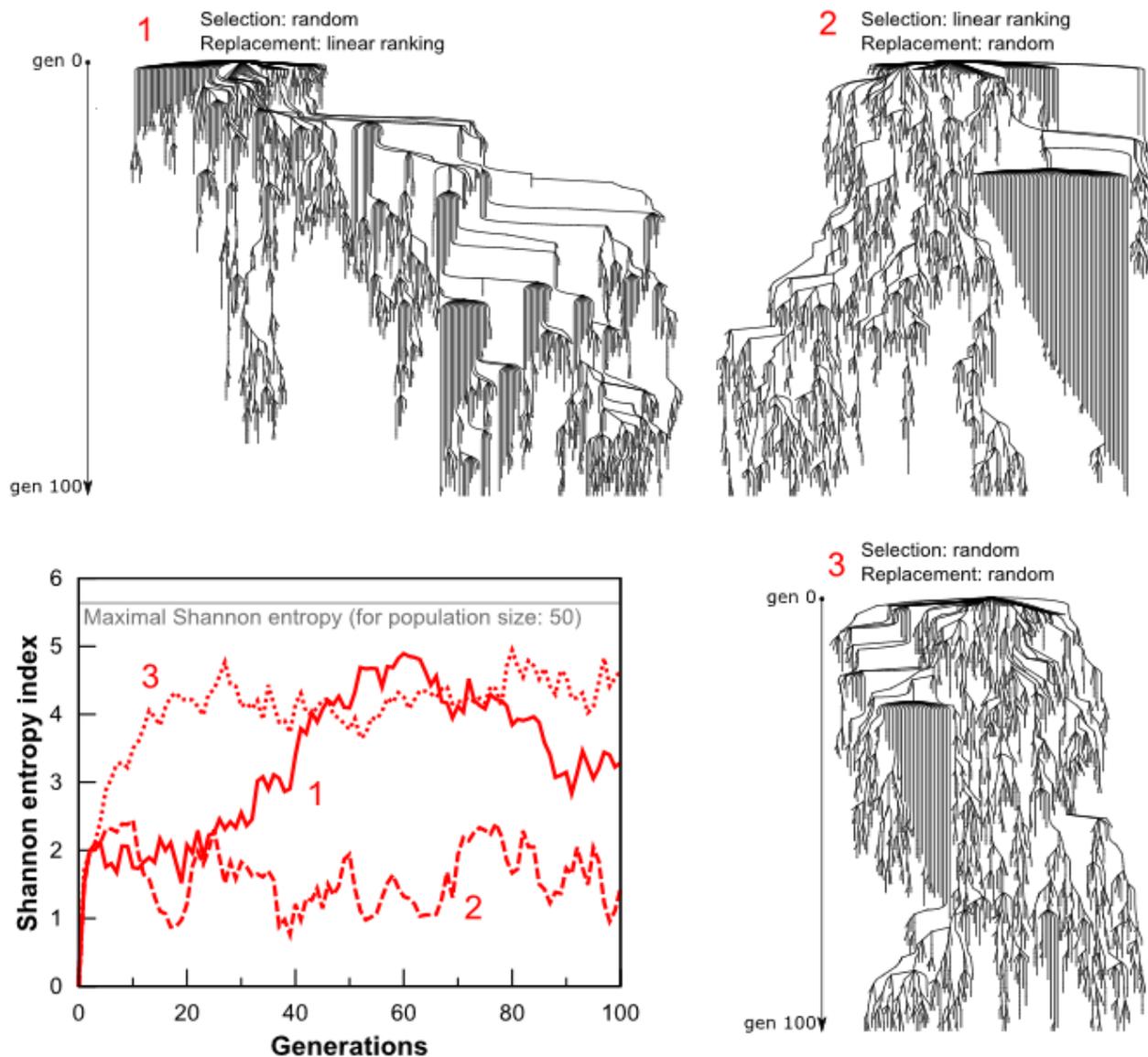


Figure 8: Example of phylogenetic trees and Shannon entropy for several selection-replacement methods during the first 100 generations. The vertical axis of phylogenetic trees represents generations, starting with the initial flowsheet down to the 100th generation, nodes represent individuals and edges ancestral relationships.

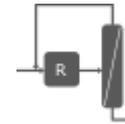
SOMMAIRE



Introduction



Approche évolutionnaire



Etude de cas



Conclusion
Perspectives pour
ces approches

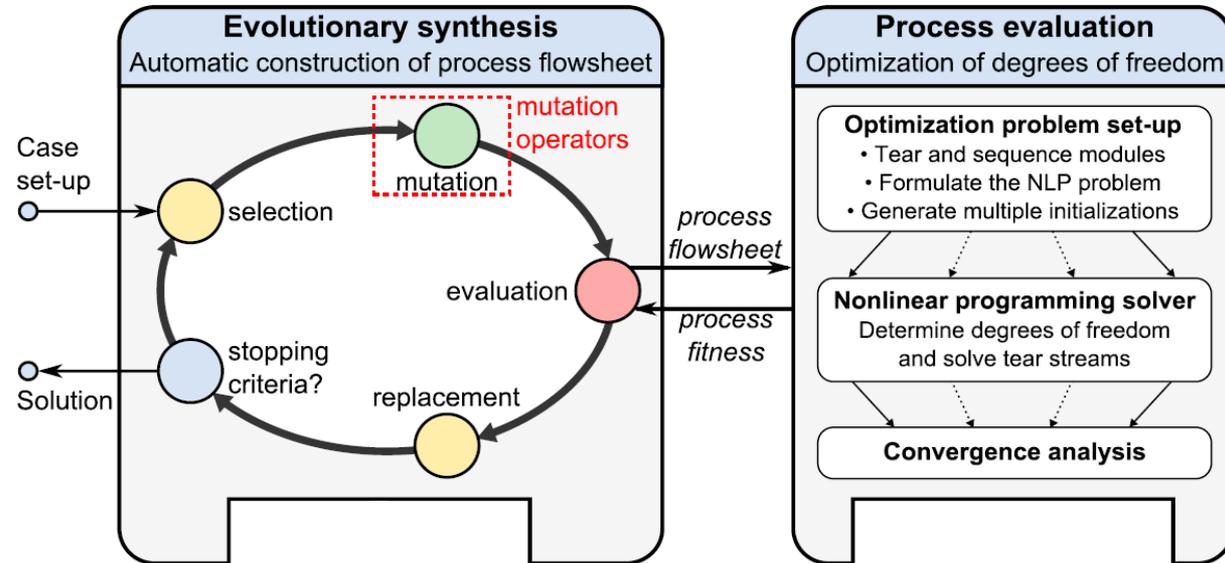


Objectif initial : générer des structures optimales (et degrés de liberté associés) **sans présupposer d'architecture**

Approche

Couplage d'une prog. évolutionnaire (dessine le procédé)

+ optim. non-linéaire (évalue la performance)



Etude de cas

- Mêmes solutions trouvées que Kokossis et Floudas
→ **proof-of-concept validée**
- Pas de nouvelle structure (car superstructure incluait 2^{32} possibilités)
- 5000 évaluations suffisent (combinatoire $> 10^9$), environ 2 min calcul



Intérêt principal de l'approche

Pas besoin de supposer un ensemble d'alternatives de procédés

→ S'affranchir d'une partie de la complexité du problème de synthèse

"Limitations arise in our ability to define an appropriate search space" (I. Grossman, 2017)

Benchmark

Cas plus complexes

- procédés très intégrés
- nombreuses opérations possibles

Intérêts comparés des approches

- quand les utiliser ?
- intérêts complémentaires ?

Drivers de la synthèse

Refléter les critères de l'ingénieur

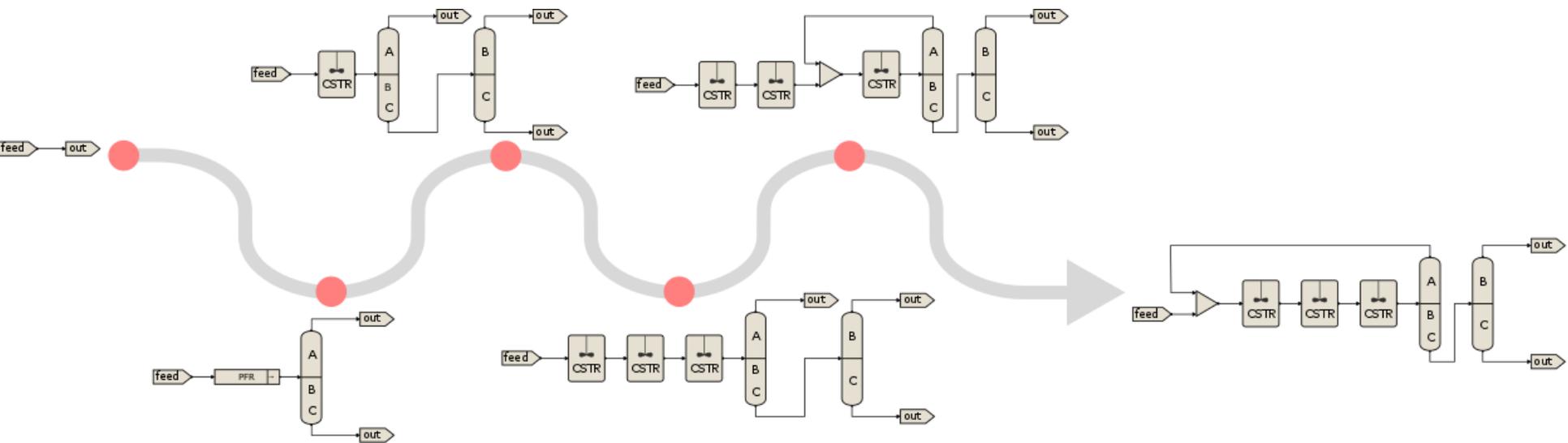
→ que veut-on obtenir ?

Introduire un critère de parcimonie

- complexité des procédés générés
- technico-économie :
chiffre certains compromis

MERCI

QUESTIONS ?



Article à venir : T. Neveux. « Ab-initio process synthesis using evolutionary programming », *under review*

RÉFÉRENCES

- Biegler (2010), "Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes", ISBN: 9780898717020*
- Cremashi (2015). "A perspective on process synthesis: Challenges and prospects", Comp. Chem. Eng. 81, pp. 130-137*
- Douglas (1988). "Conceptual design of chemical processes", John Wiley & Sons, Ltd.*
- Kokossis and Floudas (1991). "Synthesis of isothermal reactor–separator–recycle systems", Chemical Engineering Science 46, pp. 1361–1383.*
- Lipson and Pollack (2000), "Automatic design and Manufacture of Robotic Lifeforms", Nature 406, pp. 974-978*
- Floudas (1998), "Nonlinear and Mixed–Integer Optimization. Fundamentals and Applications", ISBN: 0195100565*
- Smith and Linnhoff (1988). "The design of separators in the context of overall processes", Chem. Eng. Res. Des. 66, pp. 195-228*
- Turkay and Grossman (1997), "Logic-based MINLP algorithms for the optimal synthesis of process networks", Comp. Chem. Eng. 20*
- Westerberg (2004). "A retrospective on design and process synthesis", Comp. Chem. Eng. 28, pp. 447-458*