Routledge
Taylor & Francis Group

TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
ISSN 2029-4913 print/ISSN 2029-4921 online

2012 Volume 18(1): 132–148
doi:10.3846/20294913.2012.661190

MULTI-OBJECTIVE STOCHASTIC SIMULATION-BASED OPTIMISATION APPLIED TO SUPPLY CHAIN PLANNING

Liana Napalkova[1], **Galina Merkuryeva**[2]

Department of Modelling and Simulation, Riga Technical University, LV-1658 Riga, Latvia
E-mails: [1]liana.napalkova@gmail.com; [2]galina.merkurjeva@rtu.lv (corresponding author)

Received 07 September 2010; accepted 05 September 2011

Abstract. The paper discusses the optimisation of complex management processes, which allows the reduction of investment costs by setting the optimal balance between product demand and supply. The systematisation of existing methods and algorithms that are used to optimise complex processes by linking stochastic discrete-event simulation and multi-objective optimisation is given. The two-phase optimisation method is developed based on hybrid combination of compromise programming, evolutionary computation and response surface-based methods. Approbation of the proposed method is performed on the multi-echelon supply chain planning problem that is widely distributed in industry and its solution plays a vital role in increasing the competitiveness of a company. Three scenarios are implemented to optimise supply chain tactical planning processes at the chemical manufacturing company based on using different optimisation methods and software. The numerical results prove the competitive advantages of the developed two-phase optimisation method.

Keywords: simulation optimisation, multi-objective evolutionary computation, multi-echelon supply chain, cyclic planning.

Reference to this paper should be made as follows: Napalkova, L.; Merkuryeva, G. 2012. Multi-objective stochastic simulation-based optimisation applied to supply chain planning, *Technological and Economic Development of Economy* 18(1): 132–148.

JEL Classification: C9, C61, C63, O21.

1. Introduction

Simulation-based optimisation of complex systems/processes is a central part of many scientific and technological investigations (Merkuryev *et al.* 2007). As shown in Figure 1, a general scheme of a simulation optimisation approach can be represented in the form of interaction of a stochastic discrete-event simulation model and an optimisation module. The stochastic discrete-event simulation model examines the behaviour of a system (i.e. the operation of a process) at separate and countable points of time, from one event to the next one. Accordingly, the optimisation module aims at improving simulation model responses by means of adjusting its input parameters subject to specific constraints.

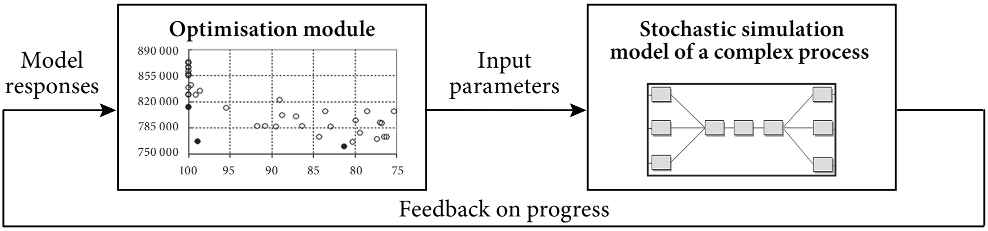

Fig. 1. General scheme of a simulation optimisation approach

A lack of a single methodology in developing simulation optimisation models has resulted in a number of optimisation methods, which are restricted to specific problems. Once more complex problems are considered, other optimisation methods should be developed. Therefore, there is a need to develop a new method that can deal with specific features of complex processes by using simulation models and is applicable to a wide range of such processes.

In this paper, the main emphasis is placed on the complex management processes, which are aimed at reduction of investment costs by setting the optimal balance between demand and supply. These processes are typical for supply chain tactical planning (SCTP) and for solving many similar optimisation problems in the separate stages of a supply chain, i.e., procurement, production, distribution and sales.

Important features of a complex management process refer to its representation as abstraction of a dynamic system that adapts to continuously changing and unpredictable environment. In general, this kind of system is characterised by the following features:

- a hierarchical network-based structure;
- a large number of decision variables including both discrete and continuous ones;
- an emergent behaviour of the system as a whole;
- a conflicting behaviour of system elements; and
- a stochastic nature.

The definition of the *hierarchical network-based structure* relates to coordination of work activities within multiple levels so that decisions at a given level depend on decisions made at upstream and downstream levels.

A large number *of continuous and discrete decision variables* results from a system size and the number of links between system elements, which results in a multi-dimensional decision search space and increase the computational complexity of the simulation optimisation problem.

The *emergent behaviour* of a system means that it cannot be predicted merely on the basis of understanding the behaviour of the system elements or from interactions between them. In opposite, all these elements working together should be investigated.

The conflicting behaviour of system elements can generate conflicting planning decisions.

Multiple performance measures of a complex system encourage searching for a set of the compromise solutions called as Pareto-optimal solutions that provide the best trade-offs for a performance of all system elements instead of a global optimal solution.

Finally, the *stochastic nature* of a system comes from the dependence of a system performance on uncertainty of the behaviour of its environment. This refers to unpredictable changes in values of environmental variables over time.

The above described features outline the class of SCTP processes that are widely distributed in industry. However, the known simulation-based optimisation methods are often unable to identify the optimal trade-offs between the conflicting objectives because of the following two major reasons:

- high computational costs required to perform simulation optimisation experiments; and
- slow convergence of multi-objective optimisation algorithms in case of mixed, i.e. continuous and discrete decision variables.

Therefore, powerful optimisation methods capable to control and optimise parameters of SCTP processes and solve the outline planning problem should be developed.

2. Optimisation problem statement

Here, the problem of optimising SCTP processes is formulated as a multi-objective and stochastic simulation-based optimisation problem with constraints and mixed decision variables:

$$\mathrm{opt}_{\mathbf{x}\in X}\ \hat{\mathbf{y}} \ = \ \mathbf{f}(\mathbf{x}) = \int_{t=1}^{T} \mathbf{y}^t \wp(\mathbf{z}^t)d\mathbf{z} = \mathbb{E}\left[\omega(\mathbf{y}^T)\right], \tag{1}$$

where $\mathbf{y}^t = \varphi(\mathbf{x}, \mathbf{c}, \mathbf{z}^t, \mathbf{s}^t)$

subject to $X: \begin{cases} \mathbf{g}(\mathbf{x}) = \mathbb{E}\left[\gamma(\mathbf{y}^T)\right] \geq 0 \\ \mathbf{h}(\mathbf{x}) = \mathbb{E}\left[\eta(\mathbf{y}^T)\right] = 0 \end{cases},$

where **f:** $X \to Y'$ is a vector of objective functions, which links values of a decision vector $\mathbf{x} = (x_1, \ldots, x_K) \in X$ with corresponding expected values of an output vector $\hat{\mathbf{y}} \in Y'$, where **x** defines a vector of K input parameters (i.e. decision variables), and $\hat{\mathbf{y}}$ defines vector of model responses (i.e. performance measures); X is a decision space; Y' is an approximate objective space; $\mathbb{E}\left[\cdot\right]$ denotesthe mathematical expectation; $\mathbf{y}^t = \left(y_1^t, \ldots, y_M^t\right) \in Y$ is an output vector at $t \in [1, T]$, where M is a number of objective functions, Y is the objective space, and T is the length of a simulation replication measured in time periods (hours, days, weeks, etc.); $\mathrm{z}^t = (\mathrm{z}_1^t, \ldots, \mathrm{z}_D^t) \in \mathrm{Z}$ is a vector of D environmental variables at $t \in [1, T]$; $\mathrm{s}^\mathrm{t} = \mathrm{s}^t = (\mathrm{s}_1^t, \ldots, \mathrm{s}_R^t) \in \mathrm{S}$ is a state vector at $t \in [1, T]$, where R is a number of state variables; **c** is a vector of model parameters, i.e., constants; $\wp$ is a probability density function of a random vector of environmental variables; φ represents a mapping that results from a simulation algorithm; **g** and **h** define vectors of inequality and equality constraints; ω, γ and η represent vectors of stochastic sample response functions.

A set of optimal trade-off solutions **x*** is composed of those potential solutions such that all components of the corresponding objective vectors cannot be simultaneously improved. This is known as the concept of *Pareto-optimality* defined as follows:

The decision vector $\mathbf{x}^* \in X$ *is said to be Pareto-optimal iff there is no* $\mathbf{x} \in X$ for which **f**(**x**) *dominates* **f**(**x***).

The concept of Pareto-optimality arises from other fundamental concept that is called the *dominance relation*. It is formally defined in the following way:
The decision vector $\mathbf{x}^* \in X$ *is said to dominate a solution* $\mathbf{x} \in X$ *if and only if* $\forall I \in \{1,\ldots,M\}$: $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})$ *and* $\exists j \in \{1,\ldots,M\}$: $f_j(\mathbf{x}^*) < f_j(\mathbf{x})$.

As it follows from this definition, a solution $\mathbf{x}^*$ is Pareto-optimal if it is not worse than any other solution for all criteria and is better for at least one criterion. For simplification, it is assumed here that all objective functions are minimised.

3. Specifying the problem requirements

Based on the state-of-the-art of simulation optimisation and the problem statement, the following requirements to the optimisation method are specified:

R1. minimise an Euclidian distance d between the true $\widetilde{\mathcal{PF}}^*$ and approximate $\widetilde{\mathcal{PF}}^*$ Pareto-optimal fronts:

$$\min d(\mathcal{PF}^*, \widetilde{\mathcal{PF}}^*);$$

R2. maximise a diversity δ^i of the Pareto-optimal solutions $\mathbf{x}^i \in \widetilde{\mathcal{PF}}^*$ to have a wide range of variety:

$$\max \sum_{i=1}^{|\widetilde{\mathcal{PF}}^*|} \delta^i;$$

R3. minimise the number ρ^τ of non-dominated solutions that are lost during the transition from iteration τ to iteration $\tau+1$:

$$\min \sum_{\tau=2}^{\tau^*} \rho^\tau;$$

R4. minimise computational costs defined as a total number of simulation optimisation iterations.

The following additional requirements are derived from the features of optimised complex processes:

R5. generate discontinuous Pareto-optimal fronts;

R6. use both continuous and discrete decision variables;

R7. include the uncertainty of system environmental variables in the search process.

It is assumed that the search space of continuous decision variables is infinite, whereas discrete decision variables are specified typically in the finite search space. This fact clearly shows that the same method cannot be equally effective for both kinds of search spaces. From this follow two additional requirements, such as:

R6.1. use global search methods for exploring the large space of discrete and continuous decision variables;

R6.2. use local search methods for exploring small regions around continuous decision variables to approach Pareto-optimal solutions as close as possible.

Thus, it becomes evident that flexible and powerful optimisation methods that are able to fulfil the above-mentioned requirements are highly valuable for efficient solving of the outline problem.

4. Overview of approaches and methods

The well-known methods and algorithms of multi-objective simulation optimisation are not remarkable for variety. Most of them use traditional techniques of aggregating multiple objectives into a single objective, or optimising the most important objective while treating other objectives as constraints. Typically, they mimic processes found in nature, but remain extremely sensitive to computational time and problem complexity. In order to relax these shortcomings and to develop a more prominent optimisation technique satisfying the requirements *R*1 ÷ *R*7, it is important to study properties of known methods and algorithms such as:

- RSM – Response surface-based method;
- SA – Stochastic approximation method;
- SPO – Sample path optimisation method;
- R&S – Ranking and selection;
- sH – Single-objective heuristics;
- mEAs – Multi-objective evolutionary algorithms.

The comparative analysis revealed that RSM, SA and SPO are used for solving optimisation problems with the continuous decision space (see Table 1). These methods require multiple initialisations with a priori preference information to approximate the Pareto-optimal front, which usually results in computationally costly search process. R&S and sH methods have more appeal for solving the outlined problem as they are able to approximate Pareto-optimal solutions by optimising both continuous and discrete decision variables. However, these advantages are often diminished through the misbalance between a high number of optimisation iterations and approximation accuracy. In contrast with the above-mentioned algorithms, mEAs directly apply the concepts of Pareto-optimality and dominance relation, and thus require only a single run for searching Pareto-optimal solutions. Most of these algorithms are able to generate discontinuous Pareto-optimal fronts by incorporating constraints handling techniques.

Table 1. Comparison of simulation optimisation methods

Requir.	Methods					
	RSM	**SA**	**SPO**	**R&S**	**sH**	**mEAs**
*R*1, *R*2	–	–	–	–	–	–
*R*3	–	–	–	–	–	+
*R*4	+	–	–	–	+	–
*R*5	–	–	–	–	–	+
*R*6	–	–	–	+	+	+
*R*7	+	+	–	–	–	–

The comparison of the simulation optimisation methods and algorithms in Table 1 demonstrates that none of them satisfies the given requirements *R*1 ÷ *R*7. It also indicates that the most rational way of solving the problem (1) is to compose a method based on determining the best hybrid combination of mEAs with other optimisation methods and algorithms.

5. Morphological analysis of multi-objective evolutionary algorithms

5.1. Parameters of hybrid algorithms

By definition, the hybridisation means the combination of the desirable properties of different methods to mitigate their individual weaknesses (Reidl 2006). This definition conforms to an idea of morphological analysis which consists in structuring and investigating a set of all possible configurations of hybrid algorithms. In this case, the morphological analysis starts from identifying the parameters of hybrid algorithms and assigning a range of relevant values to each of these parameters (see Fig. 2).

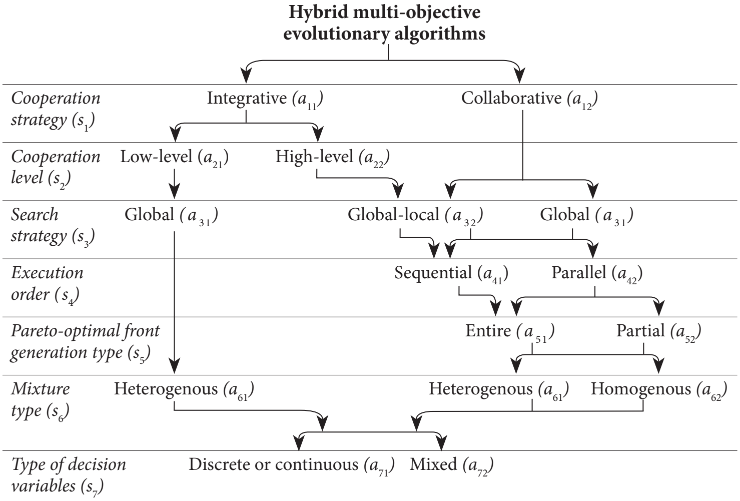

Fig. 2. The links between the parameters of hybrid algorithms

The first three parameters are identified from existing taxonomies of single-objective metaheuristics (Talbi 2002), while others are proposed for fitting the multi-objective optimisation area. Together, they provide the basis for construction of a morphological box. Detailed description of the parameters and their values can be found in Napalkova (2009). Thus, the total number of configurations in a morphological field resulted from these parameters is equal to $L = l_1 \times l_2 \ldots l_7 = 128$, where l_i is the number of relevant values for i-th parameter ($i = 1, 2, \ldots, 7$). Each configuration $A_k = (a_{1j}^{(k)}, \ldots, a_{7j}^{(k)})$ defines a certain hybrid algorithm, where $a_{ij}^{(k)}$ denotes the j-th value of i-th parameter in the k-th morphological configuration.

5.2. Selection of the morphological configuration required

In order to reduce the morphological field to a smaller set of consistent configurations and select the final configuration which satisfies the given requirements ($R_1 \div R_7$), the following existing hybrid multi-objective evolutionary algorithms are examined: (1) the genetic algorithm running on the internet (GAIN), (2) the parallel single front genetic algorithm (PSFGA), (3) the divided range multi-objective genetic algorithm (DRMOGA), (4) the parallel strength Pareto multi-objective evolutionary algorithm (PSPMEA), (5) the parallel multi-objective evolutionary algorithm with a hyper-graph represented population structure (pMOHypEA), (6) the multi-objective cellular genetic algorithm MOCell, (7) the simple multi-objective genetic local search algorithm (S-MOGLS), (8) the memetic Pareto-archived evolution strategy (M-PAES), (9) the hybrid EA (10) and hybrid NSGA-II (Stanley, Mudge 1995; de Toro *et al.* 2004; Hiroyasu *et al.* 1999; Xiong, Li 2003; Mehnen *et al.* 2004; Ishibuchi, Kaige 2004; Knowles, Corne 2000; Talbi *et al.* 2001; Deb, Goel 2001).

In accordance with the above parameters these algorithms were divided into subsets corresponding to five morphological configurations $A_1 \div A_5$.The first configuration $A_1 = (a_{12}, a_{31}, a_{42}, a_{51}, a_{62}, a_{71})$ relates to the GAIN (Stanley, Mudge 1995) and PSFGA (de Toro *et al.* 2004) algorithms and employs so called master-slave approach (see Fig. 3). A single computer (called the master) takes control of the selection for mating, whereas the other computers (called the slaves) execute fitness evaluation, crossover and mutation operators independently. The communication between computers is usually set-up through the internet. Each slave computer requires a copy of a simulation model to be installed. It may lead to the difference between execution times of the evolutionary algorithms in particular slave computers. As a result, the master computer will be forced to wait for the slowest slave computer to perform the selection operation, which can negatively affect on the execution time. In order to speed up the optimisation process, a critical number of computers with identical processing times is selected. In this case, these algorithms may outperform classical multi-objective EAs by a lower execution time. Nevertheless, because of the implementation complexity, there is a risk of getting worse results than ones previously expected.

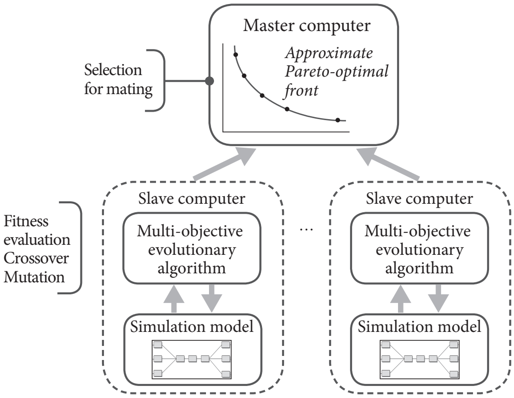

Fig. 3. General scheme of the algorithm based on A_1

The configuration $A_2 = (a_{12}, a_{31}, a_{42}, a_{52}, a_{62}, a_{71})$ lies in the kernel of algorithms denoted as DRMOGA (Hiroyasu *et al.* 1999), PSPMEA (Xiong, Li 2003) and pMOHypEA (Mehnen *et al.* 2004). In this approach, a population of candidate solutions is divided into sub-populations called the islands, each of which is associated with a particular objective function or a certain range of the Pareto-optimal front. In each island, the multi-objective EA or another heuristic method is performed for several iterations, and non-dominated solutions are copied to the external archive. Periodically, each island randomly selects some solutions from its population and sends them to the neighbouring island defined by a ring topology. This process is called the migration. After it, the worst solutions are replaced by the immigrants according to the ranking in each island, and the optimisation is continued. Finally, all Pareto-optimal solutions are collected from the islands to obtain the Pareto-optimal front. However, if this approach is applied to solving simulation optimisation problems, simultaneous running of multiple simulation models on a single computer will require powerful computers and additional parallelisation techniques (see Fig. 4).

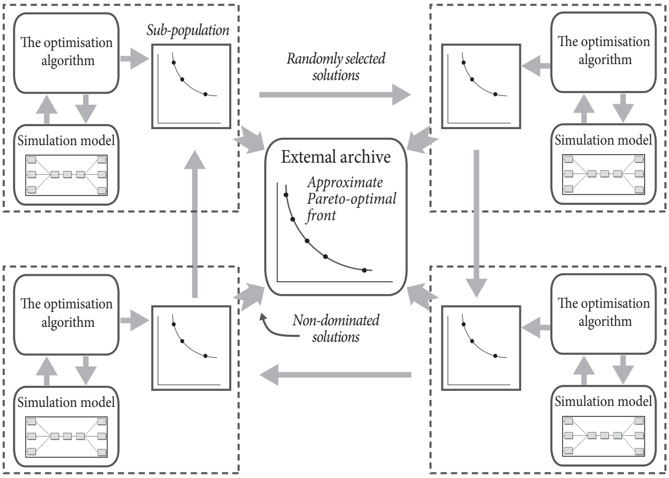
The optimisation algorithm
Sub-population
Simulation model
Randomly selected solutions
The optimisation algorithm
Simulation model
Extemal archive
Approximate Pareto-optimal front
The optimisation algorithm
Simulation model
Non-dominated solutions
The optimisation algorithm
Simulation model

Fig. 4. General scheme of the algorithm based on A_2

The configuration $A_3 = (a_{11}, a_{21}, a_{31}, a_{61}, a_{71})$ specifies recent developments concerning the replacement of computationally expensive simulation models with artificial neural networks (ANNs) (Fig. 5). The ANNs play the role of metamodels that are used to approximate black-box objective functions over the range of interest. On this way, Knowles and Corne (2000) proposed a metamodel-assisted multi-objective evolutionary algorithm, wherein the fitness of chromosomes is pre-evaluated by using a radial basis function network in order to filter the poorly performing chromosomes and direct only non-dominated ones to the exact evaluation.

However, the role undertaken by the metamodels in multi-objective optimisation is considerably harder than in single-objective optimisation. The reason is that the population

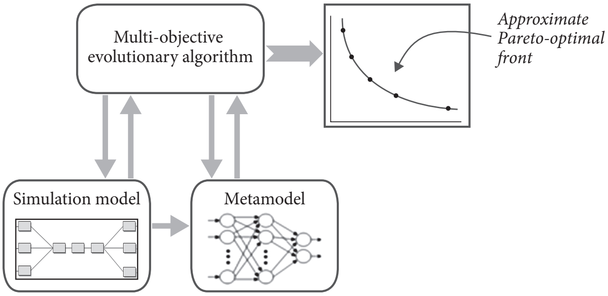
Multi-objective evolutionary algorithm
Approximate Pareto-optimal front
Simulation model
Metamodel

Fig. 5. General scheme of the algorithm based on A_3

remains spread over a relatively extended area of the search domain throughout the evolution. Additionally, metamodels require learning process and, for this purpose, an adequate set of already evaluated solutions must be available. Moreover, for ANN-based metamodels a large training data set (usually more than 100 data points) must be collected.

The configuration $A_4 = (a_{11}, a_{22}, a_{32}, a_{41}, a_{51}, a_{61}, a_{71})$ refers to genetic local search algorithms, such as S-MOGLS (Ishibuchi, Kaige 2004) and M-PAES (Knowles, Corne 2000). They are based on hybridisation of genetic operators of multi-objective EA with one of the local search algorithms. The local search can be performed by hill climbing, tabu search, simulated annealing, response surface-based methodology, etc. These methods apply the local search to every candidate solution generated by EA so that an improved solution is competing with the population for survival to the next population.

In using the S-MOGLS algorithm, the local search is probabilistically applied to candidate solutions found by NSGA-II (Deb *et al.* 2001). In this connection, multiple objective functions are aggregated by introducing weight coefficients that are randomly generated. Instead, more sophisticated the M-PAES algorithm includes two phases, such as local search phase at which a classical PAES algorithm is applied and the recombination phase. To maintain a finite set of non-dominated solutions found, a global archive is introduced, whereas a local archive is used as the comparison set in each of the local search phases.

These algorithms benefit from the overall perspective of EAs and good convergence of local search algorithms to the local optimal solutions. Such approach can significantly increase the approximation accuracy of the Pareto-optimal front, while keeping the diversity of solutions along the front. However, running the local search after each generation of EA is computationally expensive (see Fig. 6).

In case of simulation optimisation, these algorithms are able to increase the approximation accuracy of the Pareto-optimal front at relatively low computational costs, because they apply the local search only after the genetic search is completed. Besides, these algorithms don’t require multiple computers and can be easily implemented on a single computer. At the same time, they require modification for optimising mixed decision variables *(*a_{71}) and operating with stochastic objective functions.

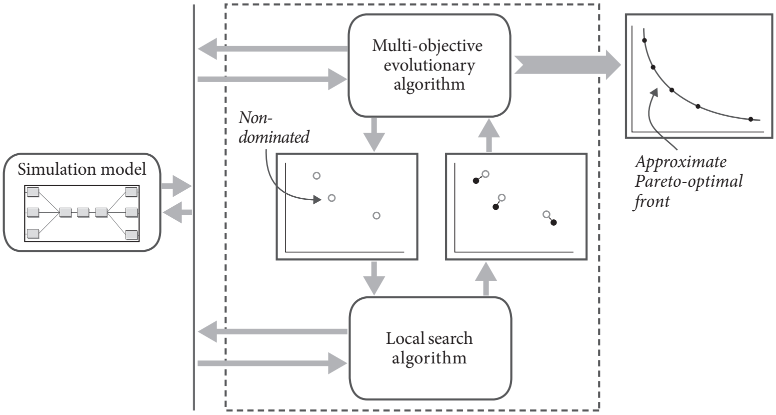
Multi-objective evolutionary algorithm
Non-dominated
Simulation model
Local search algorithm
Approximate Pareto-optimal front

Fig. 6. General scheme of the algorithm based on A_3

The configuration $A_5 = (a_{12}, a_{32}, a_{41}, a_{51}, a_{61}, a_{71})$ is associated with the hybrid EA developed by Talbi *et al.* (2001) and the hybrid NSGA-II developed by Deb and Goel (2001). Here, the difference is that the multi-objective EA is used at the first phase to obtain the approximate Pareto-optimal front and to keep the uniform distribution among the Pareto-optimal solutions, whereas the local search algorithm is aimed at improving the approximation accuracy of this front at the second phase (see Fig. 7).

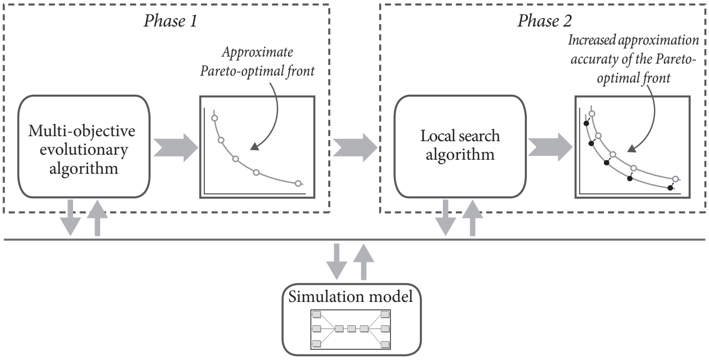
Phase 1
Approximate Pareto-optimal front
Multi-objective evolutionary algorithm
Phase 2
Increased approximation accuraty of the Pareto-optimal front
Local search algorithm
Simulation model

Fig. 7. General scheme of the algorithm based on A_5

By this reason, the configuration $A^*_5 = (a_{12}, a_{32}, a_{41}, a_{51}, a_{61}, a_{72})$ that is the nearest to the configuration A_5 is selected for the development of the hybrid algorithm required. While selecting this configuration, it is taken into account that hybrid analogues such as hybrid EA and hybrid NSGA-II have additional disadvantages as follows:

- a fixed number of optimisation iterations, which is defined as a termination criterion in the hybrid EA, doesn't permit to measure the algorithm's convergence level;
- local search algorithms used in both hybrid algorithms are not powerful enough to perform a local improvement of Pareto-optimal solutions in condition of simulation optimisation;
- local search algorithms require aggregating multiple objective functions into a weighted sum, which can cause a search in wrong directions.

Thus, the modification of the selected hybrid analogues and a removal of the above-mentioned disadvantages can provide a new framework resulting in new method of solving the problem. Below, this method is called as the two-phase optimisation (TPO) method.

6. Two-phase optimisation method

The TPO method (Merkuryeva, Napalkova 2010) simulates processes of natural evolution and linear approximation in solving the considered problem. Thus it works on a population of individuals each of which represents a search point in the space of solution candidates. Using strong simplifications the set of solution candidates is subsequently modified by means of two basic principles: *selection* and *variation*. While selection mimics the competition for reproduction among these solutions, the other principle, variation, imitates the natural capability of creating new solutions by means of crossover and mutation. Implementation of linear approximation aimed at a local search of better solutions enhances a speed and accuracy of convergence of search process to true optimal solutions.

6.1. Outline view of the TPO method

A general scheme of the TPO method is represented in Figure 8. The scheme operates starting from the global search in Phase 1 of the search stage, in which the multi-objective simulation-based genetic algorithm (MOSGA) is used to find near-optimal values of discrete and continuous decision variables in order to optimise multiple objective functions subject to specific constraints and assumptions. During the local search in Phase 2, the RSM-based linear search algorithm improves the values of continuous decision variables found in Phase 1. An output of Phase 2 is the approximate Pareto-optimal front $\widetilde{\mathcal{PF}}^*$ (its mapping into the decision space results in the Pareto-optimal set $\tilde{\mathcal{P}}^*$). In the selection stage a single Pareto-optimal solution $\mathbf{x}^*$ is determined by using the compromise programming method that measures deviations of all found Pareto-optimal solutions from an ideal (utopian) point predefined by decision maker.

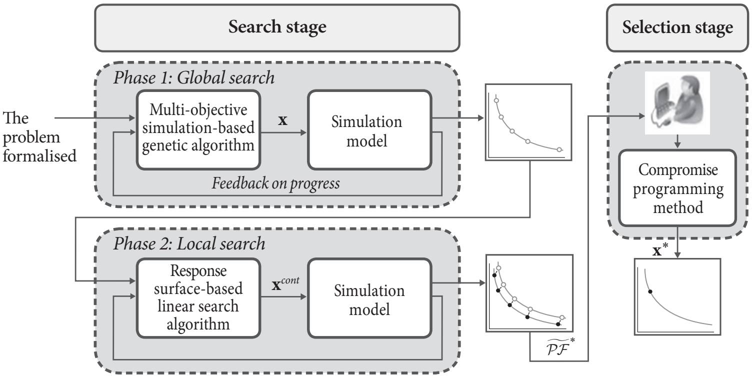

Fig. 8. General scheme of the TPO method

6.2. Formalised description of the MOSGA algorithm

Being a modified version of NSGA-II (Deb *et al.* 2001), the MOSGA algorithm (Napalkova, Merkuryeva 2008) contains both standard elements and problem-specific features to satisfy the requirements. Formally, it can be described as follows.

Let $P^{\tau}{}_{MOSGA} = \{a^{n1}, a^{n2} \mid n = \overline{1, N}\}$ be a population that consists of *N* diploid chromosomes. Each diploid chromosome in $P^{\tau}{}_{MOSGA}$ is represented by two binary strings, such as:

$$a^{n1} = \left(a^{n1}{}_{\ell_1 - 1} a^{n1}{}_{\ell_1 - 2} \ldots a^{n1}{}_1 a^{n1}{}_0\right) \in \{0,1\}^{\ell_1}, \qquad (2)$$

$$a^{n2} = \left(a^{n2}{}_{\ell_2 - 1} a^{n2}{}_{\ell_2 - 2} \ldots a^{n2}{}_1 a^{n2}{}_0\right) \in \{0,1\}^{\ell_2}, \qquad (3)$$

where ℓ_1 and ℓ_2 are lengths of strings; a_k^{n1} and a_k^{n2} are genes at locus *k*.

Binary strings a^{n1} are used in order to encode discrete decision variables $x_i^{n,discr}$ measured on a time scale by using a modified binary encoding, such as:

$$a^{n1} = \varphi(\log_2 \left(\frac{x_i^{n,discr}}{t} \right)), \tag{4}$$

where φ is an encoding algorithm; t is a basic period or the minimal available value of $x_i^{n,discr}$.

Binary strings a^{n2} are intended for continuous decision variables $x_i^{n,cont}$, and they are developed by using classical binary encoding procedure.

The performance of the MOSGA algorithm is controlled by the operator $\mathcal{G}$ that implements iterative transitions between populations according to:

$$P^{\tau_{MOSGA}+1} \sim \mathcal{G}(P^{\tau_{MOSGA}}). \tag{5}$$

This operator is comosed of four parts, such as the crowded-two tournament selection $(\mathcal{A}_s)$, the uniform crossover $(\mathcal{A}_c)$, the mutation $(\mathcal{A}_m)$ and the reproduction $(\mathcal{A}_r)$ so that:

$$\mathcal{G} = \mathcal{A}_s \circ \mathcal{A}_c \circ \mathcal{A}_m \circ \mathcal{A}_r. \tag{6}$$

The crowded-two tournament selection operator $\mathcal{A}_s$ maps the n-th string into multiple copies of itself according to its dominance depth r^n and crowding distance δ^n. The dominance depth r^n defines a dominance degree of a certain solution, where the value "1" corresponds to non-dominated solutions. The crowding distance δ^n estimates the density of solutions surrounding the n-th solution, where the value "∞" indicates the less crowded area.

The values of r^n and δ^n are estimated based on the values of performance measures $\hat{\mathbf{y}}^n$ that are obtained from simulation experiments. In order to reduce a computational time, the evaluation of solution feasibility is performed after the first simulation replication based on $f_j(\mathbf{x}^n) < \gamma_j$, where γ_j is a lower bound of the j-the objective function. If the solution $\mathbf{x}^n$ is infeasible, then simulation process is terminated.

After applying the uniform crossover $\mathcal{A}_c$ and mutation $\mathcal{A}_m$ operators, the new population $P^{\tau_{MOSGA}+1}$ is replaced by the union of the best parents $P^{\tau_{MOSGA}}$ and mating pool $\mathcal{M}^{\tau_{MOSGA}}$ to avoid the loss of non-dominated solutions during the evolution process. Dominance depths of chromosomes are updated in the combined population $P^{\tau_{MOSGA}} \cup \mathcal{M}^{\tau_{MOSGA}}$. First N solutions are gathered for the next population $P^{\tau_{MOSGA}+1}$, and so on. The MOSGA algorithm is automatically terminated, when a number τ_{MOSGA} of populations with a stagnant non-dominated set is equal to the predefined value τ_{MOSGA}^*.

Since the MOSGA is a stochastic algorithm, it could produce different solutions for different random number seeds. This is supported by performing several independent optimisation experiments based on using unique random number seeds. Then, a composite set of best non-dominated solutions is created.

6.3. Local search using the RSM-based linear search algorithm

The RSM-based linear search algorithm (Merkuryeva 2005) applied to simulation optimisation presents an iterative procedure that in each iteration includes the following steps:

– a local approximation of a response surface function;

- checking the fit of a metamodel;
- a linear search in steepest descent direction.

The algorithm starts from a local approximation of the simulation response surface by a linear regression metamodel in a small region of independent factors. In order to fit the metamodel, the Plackett-Burman experimental design is created, in which response values are received from simulation experiments.

If the metamodel is adequate, then a local response surface is sequentially investigated by using a linear search in the steepest descent direction in order to improve the values of continuous input factors. The local search is performed for input factors that correspond to significant regression coefficients. If the metamodel is not adequate or further improvement is impossible, then RSM-based linear search algorithm is terminated.

Thus, the approximation accuracy of the Pareto-optimal front $\widetilde{\mathcal{PF}}^{*}_{MOSGA}$ is increased by substituting some solutions by better ones found during the local search.

6.4. Compromise programming method for selecting a single solution

The obtained Pareto-optimal front $\widetilde{\mathcal{PF}}^{*}$ is analysed in order to select a single solution that could be most suitable for the implementation in practice. For that, compromise programming method proposed by Zeleny (1982) is used. The basic idea of this method is to identify an ideal trade-off solution, for which optimal values of objectives are usually given by decision maker. Then, the task is to find a solution that is closest to the ideal one.

There are various metrics to calculate the degree of closeness. In this work, the distance metric L_p defined as the p-root of the sum of deviations of the j-th normalised objective f_j^{norm} from its normalised ideal value z_j^{norm}, which are raised in the p power, is used:

$$L_p \colon d(\mathbf{f}^{norm}, \mathbf{z}^{norm}) = (\sum_{i=1}^{M} \left| f_i^{norm}(\mathbf{x}) - z_i^{norm} \right|^{p})^{1/p}, \tag{8}$$

where d is the distance between the ideal and Pareto-optimal solution measured on the objective space; p is a power parameter ranging from 1 to ∞**.**

7. Case study

The case study is aimed at evaluating the performance of the developed simulation-based hybrid optimisation algorithm and TPO method by solving the multi-echelon cyclic planning problem for a chemical manufacturing company (Merkuryeva *et al.* 2011). The problem is to optimise parameters of cyclic plans in order to minimise inventory holding, ordering and production costs, and maximise end-customers fill rate. These parameters include *process cycles* and *order-up-to levels,* where cycles are synchronised to schedule reorder decisions along the planning horizon, and order-up-to levels (target levels) are required to define the replenishment order quantities.

7.1. Assumptions and input data

The main operations in the manufacturing supply chain are following. First, raw materials are converted to the liquid based raisin in the plant CH. Then, they are either delivered to direct customers in Frankfurt and Pamplona or shipped to the plant DE, where other components are added to make different chemical products. At last, the end-products are shipped to direct customers connected with this plant.

The layout of supply chain simulation model is shown in Figure 9. The model is generated automatically using a simulation-based environment described in Merkuryev *et al.* (2007). The correspondent simulation model is described as follows. It is represented by 42 stages decomposed into 42 stock points (represented by triangles) and 41 processes (represented by rectangles). The end-customer demand is normally distributed; and cycles are defined according to the power-of-two policy. Cycles are presented in weeks as follows, 7, 14, 28, 56, where 56 days is the maximal cycle, which corresponds to one full turn of a 'planning wheel'. Initial stocks at end-customer echelons are equal to order-up-to levels plus average demand multiplied by cycle delays. Stock point 1 has infinite on hand stock and is not controlled by any policy. Backorders are delivered in full. The detailed description of supply chain simulation model can be found in Merkuryeva and Napalkova (2009).

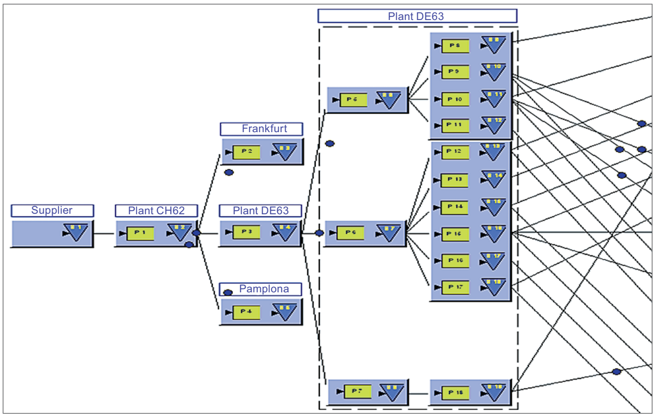

Fig. 9. Layout of supply chain simulation model

7.2. Multi-echelon cyclic planning problem statement

Two objective functions are introduced in the problem. The first one is to minimise the average total cost represented by the sum of inventory holding, production and ordering costs, i.e.:

$$\text{Min } E[TC] = \sum_{t=1}^{T}(\sum_{j=1}^{J} CP_{jt} + \sum_{i=1}^{I} CO_{it} + \sum_{i=1}^{I} CH_{it}), \tag{9}$$

where TC is the total cost, CP_{jt} is production cost in process j per period t, CO_{it} is ordering cost at stock point i per period t, and CH_{it} is inventory holding cost at stock point i per period t; I and J correspond to the number of stock points and processes, and T defines the number of periods in the planning horizon.

The second objective function is to maximise the average product fill rate:

$$\text{Max } E[FR] = E[100 * \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{k=1}^{K} QC_{ikt} \Big/ \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{k=1}^{K} D_{kit}], \tag{10}$$

where QC_{ikt} is the sum of orders delivered by stock point *i*to end-customer *k* in time period *t*, D_{kit} is actual demand of end-customer *k* to stock point *i* in period *t*. Controlled in simulation experiments, this performance measure is introduced to avoid unconstrained minimisation of the total cost.

The decision variables are lengths of cycles Cy_i and order-up-to-levels S_i, which are interpreted as discrete and continuous type variables, respectively. In the problem, specific constraints are introduced that define cycles by the power-of-two policy, in which cycles are integers and multiples of two.

7.3. Comparison of the computational costs and solutions

The case study includes three optimisation scenarios in order to evaluate computational costs in order to solve the problem using optimisation software, and the proposed method and optimisation algorithms:

- Scenario 1: SimRunner® optimisation software is applied;
- Scenario 2: OptQuest® optimisation software is applied;
- Scenario 3: TPO method is applied.

SimRunner® and OptQuest® are commercial simulation-based optimisation software, which are add-on compatible with discrete-event simulation tools. They automatically seek a course of actions to optimise performance of a simulated system by means of intelligent searching of a near-optimal solution. For this, SimRunner® combines the genetic algorithm with the evolution strategy, whereas OptQuest® is based on tabu search and scatter search. In both SimRunner® and OptQuest® software, artificial neural network-based metamodels are applied to substitute time-consuming simulation optimisation experiments in order to evaluate a model performance.

Computational costs are defined in terms of the number of optimisation iterations τ and selected as performance indicators (see Table 2). In first two optimisation scenarios, 100 and 103 iterations were required in order to find a single solution instead of the Pareto-optimal front. But, the TPO method generated the approximate Pareto-optimal front in only 67 iterations.

Table 2. Comparison of results for experiments conducted

Scenarios	Solution ID	E[*TC*], €	E[*FR*], %	τ
Scenario 1	1	904.261	86.76	100
Scenario 2	1	869.192	85.32	103
Scenario 3	1	787.431	100.00	67
	2	756.178	98.88	
	3	752.300	93.76	

Table 3. Summary of the distance values for Scenarios 1–3

Scenarios	Solutions		
	1	2	3
Scenario 1	1.036	–	–
Scenario 2	0.881	–	–
Scenario 3	0.428	0.276	0.286

The compromise programming method is used to define which of three scenarios provides the solution that is closest to the ideal solution $\mathbf{z}^{norm}$. In this case, the average total cost and average fill rate of the ideal solution $\mathbf{z}$ are defined by € 700.000 and 100%, correspondingly. As follows from Table 3, the minimal distance value $d\ (\mathbf{f}^{norm},\mathbf{z}^{norm}) = 0.276$ is provided by the second solution of the TPO method.

8. Conclusions

The paper investigated multi-objective simulation-based optimisation methods and algorithms that can be applied to solving the multi-objective constrained and stochastic simulation-based optimisation problem. Analysis showed that multi-objective evolutionary algorithms fulfil the most of requirements imposed on the improved search of Pareto-optimal solutions. However, these algorithms, cannot simultaneously ensure high approximation accuracy and diversity of the Pareto-optimal front, and require high computational costs to generate the front. Therefore, morphological analysis of hybrid multi-objective evolutionary algorithms was conducted to determine the best combination of parameters that satisfy the formulated problem requirements.

The TPO method was developed based on using the hybrid optimisation scheme. In the method, multi-objective simulation-based genetic and response surface-based linear search algorithms were sequentially used to perform the global and the local search while optimising discrete and continuous parameters of supply chain cyclic plans. In order to prove the competitive advantages of the proposed TPO method, a case study with three optimisation scenarios was performed. The results showed increase of approximation accuracy of the Pareto-optimal front, while decreasing a number of simulation optimisation iterations. Further research on multi-objective simulation-based optimisation of complex processes will include application of the proposed method to a wider range of problems related to the optimisation of complex management processes.

References

de Toro, F.; Ortega, J.; Ros, E.; Mota, S.; Paechter, B.; Martin, J. 2004. PSFGA: parallel processing and evolutionary computation for multiobjective optimisation, *Parallel Computing* 30(6): 721–739. http://dx.doi.org/10.1016/j.parco.2003.12.012

Deb, K.; Goel, T. 2001. Controlled elitist non-dominated sorting genetic algorithms for better convergence, in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimisation*, 67–81. http://dx.doi.org/10.1007/3-540-44719-9_5

Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T. 2001. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II, in *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 849–858.

Hiroyasu, T.; Miki, M.; Watanabe, S. 1999. Divided range genetic algorithms in multi objective optimisation problems, in *Proceedings of International Workshop on Emergent Synthesis*, 57–66.

Ishibuchi, H.; Kaige, S. 2004. Implementation of simple multiobjective memetic algorithms and its application to knapsack problems, *International Journal of Hybrid Intelligent Systems* 1(1–2): 22–35.

Knowles, J. D.; Corne, D. W. 2000. M-PAES: a memetic algorithm for multiobjective optimisation, in *Proceedings of the 2000 Congress on Evolutionary Computation*, 325–332.

Mehnen, J.; Michelitsch, T.; Schmitt, K.; Kohlen, T. 2004. pMOHypEA: parallel evolutionary multiobjective optimisation using hypergraphs, *Technical Report* No. CI-189/04.

Merkuryev, Y.; Merkuryeva, G.; Desmet, B.; Jacquet-Lagrèze, E. 2007. Integrating analytical and simulation techniques in multi-echelon cyclic planning, in *Proceedings of the First Asia International Conference on Modelling and Simulation*, 460–464. http://dx.doi.org/10.1109/AMS.2007.54

Merkuryeva, G. 2005. *Response Surface-based Simulation Metamodelling Methods with Applications to Optimisation Problems*. Supply chain optimisation Product / Process Design, Facility Location and Flow control. Springer, 205–215. http://dx.doi.org/10.1504/IJSPM.2010.032654

Merkuryeva, G.; Merkuryev, Y.; Vanmaele, H. 2011. Simulation-based planning and optimization in multi-echelon supply chains, *Simulation: Transactions of the Society for Modeling and Simulation International* 87(8): 698–713.

Merkuryeva, G.; Napalkova, L. 2009. *Supply Chain Cyclic Planning and Optimisation*. Simulation-Based Case Studies in Logistics: Education and Applied Research. London: Springer-Verlag, 89–107.

Merkuryeva, G.; Napalkova, L. 2010. Two-phase simulation optimisation algorithm with applications to multi-echelon cyclic planning, *International Journal of Simulation and Process Modelling (IJSPM)* 6(1): 7–18.

Napalkova, L. 2009. Hybridisation of evolutionary algorithms for solving multi-objective simulation optimisation problems, in *RTU 50th International Scientific Conference*, 9–15.

Napalkova, L.; Merkuryeva, G. 2008. Theoretical framework of multi-objective simulation-based genetic algorithm for supply chain cyclic planning and optimisation, in *Proceedings of the 10th International Conference on Computer Modelling and Simulation (EUROSIM/UKSim'2008)*, 467–474.

Reidl, G. 2006. A unified view on hybrid metaheuristics, in *Proceedings of Hybrid Metaheuristics*, 1–12.

Stanley, T.; Mudge, T. 1995. A parallel genetic algorithm for multiobjective microprocessor design, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 597–607.

Talbi, E.-G. 2002. A taxonomy of hybrid metaheuristics, *Journal of Heuristics* 8(5): 541–564. http://dx.doi.org/10.1023/A:1016540724870

Talbi, E.-G.; Rahoual, M.; Mabed, M. H.; Dhaenens, C. 2001. A hybrid evolutionary approach for multicriteria optimisation problems: application to the flow shop, in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimisation*, 416–428. http://dx.doi.org/10.1007/3-540-44719-9_29

Xiong, S.; Li, F. 2003. Parallel strength Pareto multiobjective evolutionary algorithm, in *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 681–683. http://dx.doi.org/10.1109/PDCAT.2003.1236390

Zeleny, M. 1982. *Multiple Criteria Decision Making*. McGraw Hill, NJ.

Liana NAPALKOVA holds Doctor of Engineering (Dr. sc. ing) from Riga Technical University (Latvia). Her research interests focus on the development of simulation-based optimisation techniques for providing competitive advantages in multi-echelon supply chain cyclic planning.

Galina MERKURYEVA is Professor at the Department of Modelling and Simulation, Riga Technical University (Latvia). She holds Doctor of Sciences (Dr Habil) from Institute of Control Sciences of Russian Academy of Sciences, and Doctor of Engineering (Dr. sc. ing) from Institute of Electronics and Computer Techniques of Latvian Academy of Sciences. Her professional interests and experiences are in areas of discrete-event simulation, simulation metamodelling and optimisation, decision support systems, supply chain simulation and simulation-based training.