Multi-criteria optimization of service productivity using evolutionary algorithm

Sven Tackenberg, Peter Schmitz, Sönke Duckwitz, Christopher M. Schlick

Institute of Industrial Engineering and Ergonomics, RWTH Aachen

The research presented is concerned with the planning component of service management and describes the use of a multiobjective evolutionary heuristic to optimize the productivity of service projects. Adequate planning assures the effective and efficient use of resources, customer satisfaction and the attainment of service objectives in an acceptable period of time. In order to achieve improved service productivity, the concepts for planning are reviewed and an optimization algorithm for a resource constrained service model is introduced. Thereby, our vision is a supporting system for planers using a multi-criteria optimization algorithm (SMS-EMOA) to reduce the risk of service development. Therefore, our approach integrates the bounded rational decision making of actors and leads to a priori optimal adjustment of relevant factors like the interaction between customers and the service provider.

1. Introduction

Several theories of service management have been published. While relevant authors fail to agree on a single theory of developing optimal service processes, six facts in terms of service processes are generally regarded as important:

- Uncertainties regarding the duration of activities are a typical characteristic.
- Predecessor constraints of activities are not strictly specified and are therefore often refined with respect to the situation and the person.
- The assignment of persons and resources (tools, facilities) to tasks has to consider different skills of the workforce and a level of ability and characteristic for each of these skills.
- Iterations are a characteristic of complex services as well as a major source of unexpected rework and budget overruns.
- The dynamic of a service process results from the bounded rational decision making of actors and the follow-up actions. This covers communication and cooperation processes between service providers and customers.
- The characteristic of a service process directly determines the productivity of a service.

We assume that a systematic planning can improve service processes. Therefore, a prospective correct description (plan) of an optimal service provision is an important contribution to enhancing service productivity. Only a few detailed models can be found in literature that explain the complexity and multidimensionality of service productivity (Grönroos; Ojasalo, 2004; Lasshof, 2006). However, all these models are very generic and limited with regard to their applicability by service managers. Thus, methods for an optimal scheduling of service processes and their application to real service processes are missing. Real-world scheduling problems during the development of services are generally complex, large scale, constrained, and multiobjective. Classical operational research techniques are often inadequate if complex aspects of a work organization are considered. With the new research on Multi Objective Evolutionary Algorithms (Tan et al., 2005; Zitzler et al., 2004; Dahal et al., 2007) there are new heuristics available to solve detailed work process models. These MOEAs are inspired from biological concepts and are highly scalable and flexible in terms of considering model constraints and multiple objectives.

2. Developing plans for services to improve productivity

We focus on a simulation based approach to measure and improve key figures of service productivity during the planning phase of a service. The approach has to be suitable for the decision support in the context of service design and should consider the main aspects of services: immateriality and interactivity. The analysis of relevant literature shows that there are many theoretical accounts of planning in the broad field of psychology. Miller et al. (1960) define planning as an execution of a behavior that matches a scheme, whereas Hayes-Roth and Hayes-Roth (1979) include anticipating a course of action as well. Funke and Glodowski (1990) define planning as the design of a sequence of actions, which can be regarded on different levels of resolution, while taking into account restricting constraints. Funke’s and Glodowski’s (1990) nomenclature serves here as the theoretical basis of a decision-support method for the development of a service plan. The authors distinguish between plan establishment and plan execution and identify several basic competencies for both phases. In order to realize the objective of a more efficient solution of a Resource Constraint Service Scheduling Problem (RCSSP) only the outcome of the plan establishment phase is of interest. The plan establishment contains the anticipatory ordering of partial steps that lie in the future, while taking into account constraints and involving memory contents. Funke and Glodowski define five competencies required for establishing a plan (Funke; Glodowski, 1990):

- identifying temporal sequences of subtasks,
- identifying spatial, temporal, material and logical conditions,
- determining subgoals: ability to divide the whole plan into several parts
- access to alternatives: planners should, if necessary, be able to generate alternative solutions,
- appropriateness of the plan: subgoals should be generated in a real-world-manageable manner.

Therefore, a method for the plan establishment phase has to support these five competencies. To evaluate the quality of a developed service organization, it is suitable to analyze the result of the planning phase (plan) instead of the underlying planning process. We define the dimensions of plan efficiency and plan effectiveness to measure the quality of a service plan. Plan efficiency describes the extent to which the planning objectives of a specific service planning task are met if the plan is exactly executed. We assume that an improved service productivity can be achieved through plan effectiveness and efficiency. Therefore, an evolutionary multiobjective algorithm is developed which is desirable to optimize the factors of plan effectiveness and efficiency. The concept of plan effectiveness measures the four aspects:

- correct depiction of activity durations,
- correct depiction of activity costs,
- correct predecessor-successor relations, and
- respecting time-, skill and resource constraints.

3. Basic principles of evolutionary optimization

Developing a detailed plan for a service provision involves an arbitrary optimization problem with *k* objectives which are all to be minimized and equally important. Therefore, every objective of a plan can be transferred to a minimization problem. Such an optimization problem is described by a decision vector $(x_1,\ x_2, \ldots, x_n)$ in a decision space *X* and an objective vector $(y_1,\ y_2, \ldots, y_n)$ in the objective space *Y*. The entries of the decision vector comprise the factors which define and parameterize a service process. A function $f\colon X \rightarrow Y$ evaluates the solution generated by a specific decision vector regarding the different objectives (fitness) of planning (Zitzler et al., 2004).

The evaluation function for the considered planning problem with multiple objectives is complex. Based on the well-known concept of Pareto dominance, an objective vector $y^{(1)}$ dominates another vector $y^{(2)}$ if no component of $y^{(1)}$ is greater than the corresponding component of $y^{(2)}$ and at least one value of $y^{(1)}$ is smaller: $\left(y^{(1)} \succ y^{(2)}\right)$. The same also applies for the decision space $\left(x^{(1)} \succ x^{(2)}\right)$, $x^{(1)}$ leads to a higher quality and therefore $x^{(1)}$ dominates $x^{(2)}$, if $f\left(x^{(1)}\right)$ dominates $f\left(x^{(2)}\right)$ (Zitzler et al., 2004). For a service planning problem there may exist several optimal objective vectors representing different trade-offs between objectives. The set of optimal decision vectors is in general named as the Pareto set $X^* \subseteq X$. The image of all vectors in $X^*$is the Pareto front $Y^* = f(X^*) \subseteq Y$. Every vector in X^* describes an optimal plan for the scheduling problem. Thereby, several optimal (non-dominated) plans can exist due to conflicting objectives. All these plans are elements of the best front (cf. Fig. 1). By analogy to natural evolution, one solution (e.g. a plan for a service provision) is called an individual and the set of individuals is called population.

Although there are several approaches to solve a multiobjective optimization problem, e.g. by aggregating the different objectives into a single one, most work in MOEAs is focused on the approximation of the Pareto front (Zitzler et al., 2004). We assume that the identification of all non-dominated solutions is the most promising approach for the service-based scheduling problem.

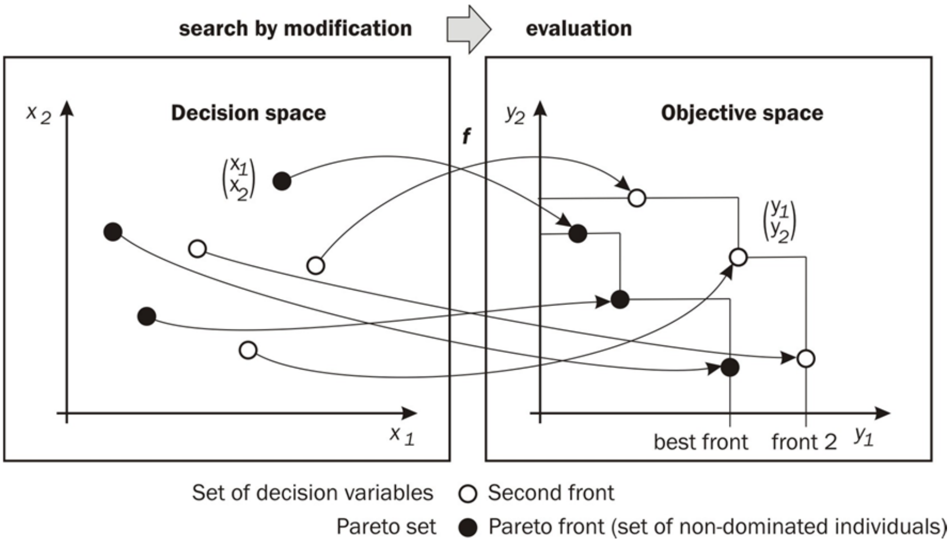

Fig. 1: Multiobjective optimization problem (Zitzler et al., 2004)

MOEAs for the approximation of the Pareto front use a stochastic search procedure. They do not guarantee the identification of all non-dominated solutions but such an algorithm try to identify a good approximation, i.e., a set of solutions whose objective vectors are (hopefully) not too far away from the optimal objective vectors.

A flowchart of a general MOEA based on the work by Tan et al. (Tan et al., 2005) is illustrated in Fig. 2. At the start, a population of solutions (individuals) is initialized and evaluated according to the user-defined objective vector $(y_1, y_2, \ldots, y_n)$. Based on these objectives, the individuals are evaluated according to dominance relations. The quality of a solution is further refined by techniques to evaluate the density relations of the identified solutions in order to distribute the non-dominated solutions uniformly along the Pareto-front. Until the quality of the solutions is insufficient and the stopping criterion is not met, genetic operations like recombination and mutation modifies the decision vector variables of selected individuals. The recombination operator stochastically selects a certain number of individuals and creates a predefined number of new individuals by combining parts of the selected solutions. By contrast, the mutation operator modifies individuals by stochastically changing a few decision variables of an individual. In literature there are several techniques for selection, mutation and recombination mentioned and evaluated regarding their performance (Murata et al., 1996). These techniques can be adapted to the specific evolutionary encoding of a service based scheduling problem.The genetic operators generate solutions for a new population (new generation *t*+1). The newly evolved population is combined with the non-dominated individuals preserved from the previous generation *t*. The combined population is evaluated according to dominance relations. If the predefined population size is exceeded by the number of non-dominated individuals, solutions are discarded which are in crowded areas of the Pareto-front. This allows a distribution of the individuals uniformly along the discovered front. Otherwise, non-dominated solutions are transferred to the population of *t*+1according to their rank. All other solutions are deleted. The remaining non-dominated and dominated solutions form the new population of generation t+1 and the evolutionary cycle is repeated until the stopping criterion is met.

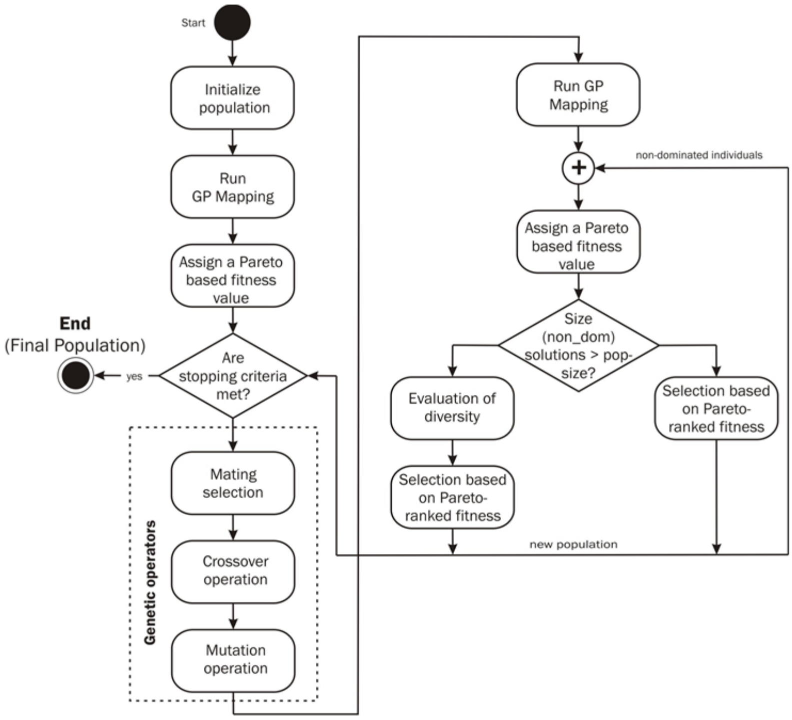

Fig. 2: Flowchart of a Multiobjective Evolutionary Algorithm (MOEA)

4. Semi formal and mathematical model of a service

4.1. Semi formal model

The prospective optimization of a service process has to be suitable for service managers. Therefore, work organizations and service processes should be easily and quickly modeled. To realize a service model with appropriate and adequate information the C3 modeling method for weakly structured and domain-spanning workflows has been extended. C3 is based on activity diagrams of the Unified Modeling Language (UML) and is tailored towards special requirements of service managers (Killich et al., 1999). The term “C3” is derived from the initial letters in the words cooperation, coordination and communication – the core elements of distributed service processes. Due to its semi-formal semantics, C3 is particularly suited for the analysis of creative processes that are associated with uncertainty and iterations. To point out the level of detail of a service model used by the developed MOEA a C3 model with the extended notation is shown in Fig. 7.

The service organization (Fig. 7) is characterized by a high degree of freedom regarding predecessor constraints of activities, the assignment of actors to activities and alternative routing. In order to complete an activity, it has to be processed by at least one person of the corresponding organizational unit which fulfils the skill requirements. Furthermore one task may lead to different valid activity characteristics (modes). The different modes can result from heterogeneous skill levels of the staff which have an impact on the processing strategy, the makespan or the iteration probability. The predecessor relations between activities are expressed by a downstream edge (control flow). Thereby, the maximum permitted overlapping is given by a value labeling the control flow. Iterations are described by a rear-oriented control flow with a given stochastic value (Activity 2: 80%, Activity 3 and 4 to Activity 2: 80%). The effort for a re-execution may differ from the first execution and is given as a percentage value of the original estimated makespan (Activity 2: 50% = 90 min. for the first iteration of *A*). After processing tasks anew, the probability of a further iterative execution is reduced by the value given in the tables for *A* and *B*. Due to the appearance of nested iterations (*B*) the triggering event is considered during the calculation of the appearance of iteration *A* and *B* as well as the duration of activity 2. The elements for the parameterization of an activity, such as minimum requirements concerning persons, resources and information as well as the estimated effort, are essential sub-elements of a C3 model. The introduced notation elements allow a holistic semi-formal description of the relations between elements and are specified by the following connectors:

- Control flow: Predetermined predecessor – successor restrictions of activities
- Information flow: The flow of information describes the emitter, the receiver and the type of information.
- Synchronization node (Branch, Merge): The Branch node describes the branching of activity sequences, which can be executed simultaneously. All activities upstream by a merge node have to be sufficiently executed (meeting the overlapping restrictions) to generate the feasibility of their successors.
- Decision node (Branch, Merge): The Branch node represents the selection of *k* activities among *n* activities. A XOR logic is used in general and describes the selection of only one successor activity or one path of activities. Therefore, only one predecessor of a merge node has to be sufficiently executed to start the successors of the node.

Service models are developed with the help of the software system *C3 editor*. A saved C3-model is used as input of the MOEA.

4.2. Formal Model

In the following sections we focus on a mathematical description of the dynamic of a service process. We assume that the dynamic of a service provision results from the decision making of the involved actors. As a consequence, the formal description of an actor, an activity and an activity net is presented. The formal specification of an activity as well as the relation between activities can be achieved by the Petri net notation (Winkelmann; Luczak, 2006).

4.2.1. Actor

We assume that a set $AP = \{1, \dots, p\}$ of persons to perform the activities is given. Every person has at least one qualification $q \in Q$ and one skill $k \in K$:

$$\sum_{q=1}^{Q} qr_{pq} \geq 1 \quad \forall\, p \in AP, q \in Q$$

$$\sum_{k=1}^{K} qr_{pk} \geq 1 \quad \forall\, p \in AP, k \in K$$

If a person $p \in AP$ *has the qualification* $q \in Q$ *the binary variable* qr_{pq} is “1”. The same also applies to qr_{pk}. The skills of a person are expressed with skill vectors and skill matrices (Firat; Hurkens, 2010). The entries of the vector describe the skill level of a person. Skill matrix entries specify the relation between a particular skill domain and a skill level of a person. As a consequence, if a person has an entry in the matrix, then the person is also qualified at lower levels in the same skill:

$$SV_p \in \{0, 1, \dots |\mathbb{L}|\}^{\mathbb{K}}$$

$$SM_p^{(l,k)} = \begin{cases} 1, & if\ l \leq SV_p^{(k)} \\ 0, & otherwise \end{cases}, \qquad SM_p \in \{0,1\}^{\mathbb{L}\times\mathbb{K}}\ \forall\, k \in K$$

A person *p* with the skill vector $SV_p = (2,0,1)$ has the following matrix *SM*:

$$SM_p = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

If an activity execution requires more than one person the matrices of the persons are added to describe the common skill level. For example, $SV_p = (2,0,1)$, $SV_m = (2,1,2)$ result in:

$$SM_{p+m} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

The working time of a person is defined as follows:

$$ap_{p,day}(t) = \begin{cases} 1, & if\ t_{AA} \leq t \leq t_{PA} \\ 1, & if\ t_{PE} \leq t \leq t_{AE} \\ 0, & otherwise \end{cases}$$

A person can only execute an activity between the start of a work day t_{AA} and lunchtime t_{PA} as well as between the end of the break t_{PE} and the start of the leisure-time t_{AE}.

A task is assigned to a person when the conditions of a sufficient execution of the predecessor activities are fully met. Afterwards, the task is shown up in the *task pool of this person*. As a result a person’s task pool can contain a number of tasks with varying processing statuses due to overlapping and uncoupled activities. Also the model takes into consideration that a person can interrupt the execution of an activity and start another one. But a person can only execute one activity at the same time. If several tasks in the pool, the person has to organize them.

Thereby, a person does not always make rational decisions. Persons are prone to seeing short-term tasks as more important than long-term ones due to the operational day-to-day business in an organization. A higher priority results only when the time frame until desired task completion continues to greatly decrease. This behavior is referred to in literature as bounded rational behavior. In order to take this behavior into consideration, the time factor must be included in a prioritization algorithm. The Temporal Motivational Theory (*TMT*) of Steel and König (Steel; Koenig, 2006) manages to do so. The prioritization algorithm of the service model is based on the findings of *TMT*. The priority that a person assigns to a task consists of several aspects.

First, each task receives a value for "importance" that results from the significance of the task in question for the particular service (I_s), the project's contribution to the organization's success (I_c) and the importance for the worker doing the processing (I_w). The importance of the task is represented by the positive effect of a task selection. At the same time, the temporal aspect during the priority calculation is also considered. The positive effect of task preparation is realized when the activity necessary for solving the task is carried out by the task's particular deadline t_{dead}, at the latest. In addition to the time span until the deadline, the makespan still needed for the task must also be considered since the urgency of a task is significantly determined by the task's already attained degree of processing. The urgency of a task or activity at time *t* results from the quotients of the work time that must still be invested and the time remaining until the deadline. The remaining processing time can then be calculated by the expected activity duration d_{Ai}, and δ_{ci}, the already reached degree of processing:

$$Urgeny = \frac{d_{Ai}(1-\delta_{ci})}{1+\Gamma^{+}(t_{dead}-t)}$$

The negative aspect of a task selection exists in the necessary familiarization (necessary training period) with the task. This negative utility is determined through a specified familiarization, d_{STi}, and the person's level of familiarization, δ_{STi}. The level of familiarization increases during the execution of activities and then decreases again during the processing breaks. This is because familiarization with the task is once again necessary for the processing of a new activity, depending on the length of the processing break. Therefore, the priority of a task in this context can be calculated as follows:

$$Priority = \frac{I_w K_w + I_c K_c + I_s K_s}{1+ \Gamma^{+}\left(\frac{T_{dead}-t}{d_{Ai}(1-\delta_{ci})}\right)} - (k_1(1-\delta_{STi})d_{STi}K_{ST} + k_2 K_{KD} D_{KD})$$

The individual weighting of the single factors is described by the values K_s, K_c, K_a and K_{ST}. Γ^{+} represents the weighting of a positive yields through the person. k_1 *and* k_2 define the weighting of the portion of costs. The person-specific definition of decision preferences is described by the individual weighting factors of the subaspects contained in the priority calculation.

4.2.2. Activity

Many previous studies have pointed out that a service process entails heterogeneous activities with complex relations. Some studies have classified the relations types of services. However, there is a limitation of a mathematical description of the relations types to identify the decision vector $(x_1, x_2, \ldots, x_n)$ of a service model. Furthermore, a systematic classification of the relations between activities is missing. To cope with these limitations, we classify the relation types of a service process according to: *de-*

pendency, *structure of activity execution*, *relation cardinality*, *overlapping, routing* and *routing characteristic and synchronization* (cf. Fig. 3).

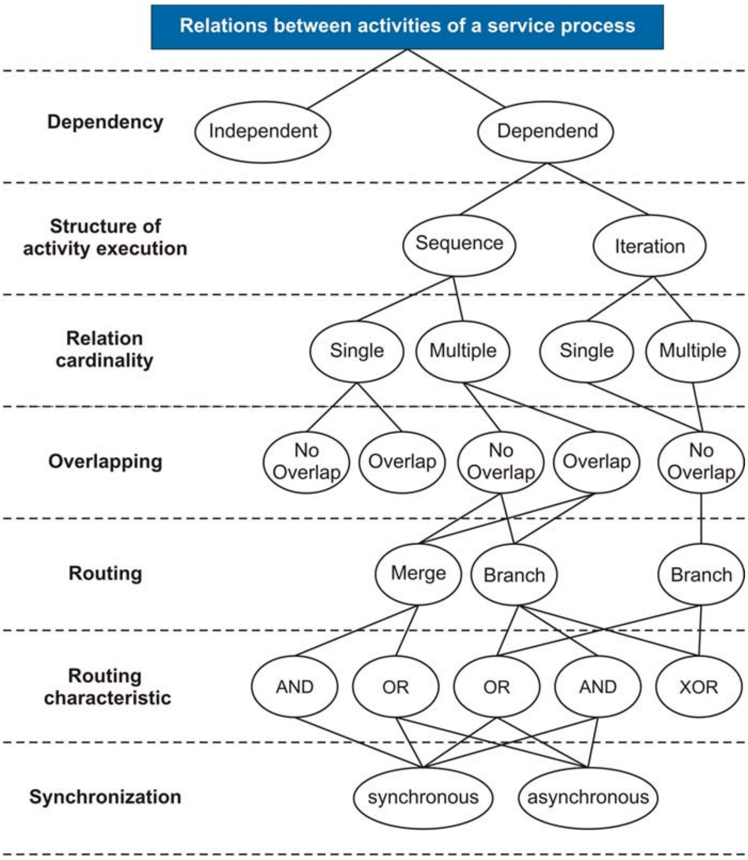

Fig. 3: Relations of activities in a service process

The central part of the model is the formal description of an activity execution. The Petri net notation offers us the opportunity to take the dynamic of such processes into account (Girault; Valk, 2003).

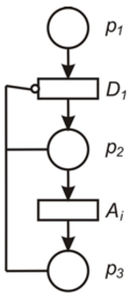

Fig. 4: Petri net of an activity

Without considering persons and resources an activity is defined as:

$$PN = (P, T, F, V, L, M_0, G);\ P = (p_1, p_2, p_3);\ T = T' = (D_1, A_i)$$

The network comprises the connecting edges $V(\ )$ and the deceleration edges $L(\ )$:

$$V(p_1, D_1) = 1;\ V(D_1, p_2) = 1;\ V(p_2, A_i) = 1;\ V(A_i, p_3) = 1$$

$$L(p_2, D_1) = 1;\ L(p_3, D_1) = 1$$

M_0 is a marking of PN and describes the initializing time point: $M_0 = (1,0,0)$

In the initial state M_0 is a token (marker) in the place p_1 which activates the transition D_1 and triggers the decrement of a timer value. This time value determines the firing of the transition D_1 and therefore the starting point of an activity. A token is set in p_2 and due to the backward oriented arc $L(p_2, D_1)$ a re-firing of the activity (activation of D_1) is disabled until the token in p_2 is removed. Furthermore, the token in place p_2 leads to an activation of the transition A_i. The time of the activation of A_i corresponds to the execution time of the activity. After expiration of the activity duration d_{Ai} the transition fires and a token appears in p_3.

$$M_0 = (1,0,0),\ \ M_1 = (0,1,0),\ \ M_2 = (0,0,1).$$

The amount of all activated transitions $T(M)$ within the marker M and the valid timer $C(M)$ are defined as:

$$T(M_0) = D_1,\ \ T(M_1) = \{A_i\},\ \ T(M_2) = \emptyset,$$

$$C(M_0) = \{D_1, A_i\} = C(M_1) = C(M_2)$$

The execution of an activity is specified by the firing of a transition and a state change of markers $M \rightarrow M'$. Therefore the state change $M_0 \rightarrow M_1$ and all new activated transitions at M_1 and M_2 are defined as:

$$N(M_1; M_0, \hat{T} = D_1) = (T(M_1)\backslash T(M_0)\backslash\hat{T})) \cap C(M_1)$$

$$= (\{A_i\}\backslash (\{D_1\}\backslash\{D_1\})) \cap \{D_1,A_i\}$$

$$= \{A_i\}$$

$$N(M_2; M_1, \hat{T} = Ai) = \emptyset$$

The amount of active transitions at M_1 which were also active at M_0 is described by:

$$O(M_1; M_0, \hat{T} = D_1) = (T(M_1) \cap (T(M_0)\backslash \hat{T})) \cap (\hat{T}\backslash C(M_1))$$

$$= (\{A_i\} \cap (\{D_1\}\backslash D_1)) \cap \emptyset = \emptyset$$

$$O(M_2; M_1, \hat{T}, = A_i) = \emptyset,$$

$$O^s(M_1; M_0; \hat{T}, = D_1)= O^s(M_2, M_1,A_i) = \emptyset.$$

The duration between activation and firing represents the starting time (transition D_1) and the makespan of an activity (transition A_i). It can therefore be deduced that the decision variables of an activity are only the activation times d_{D_i} and d_{A_i} which have to be optimally adjusted by a MOEA. Precedence relations between activities enforce that all predecessors must be sufficiently completed before an activity can start. This restriction is represented by setting valid values for d_{D_i} of all successors of an activity. For all relation types in Fig. 4 Petri net models are defined in the same way (cf. Fig. 5). This allows the derivation of the decision variables which specifies the structure of a service process.

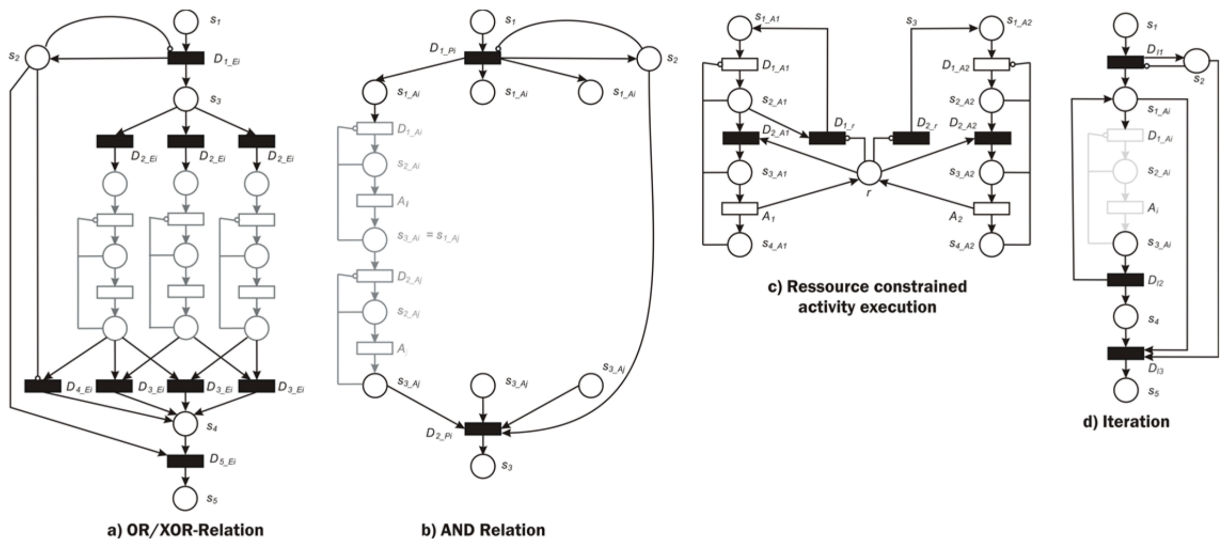

Fig. 5: Petri net models of activity relations

An activity requires for processing at least one qualification and one specific skill of a person. The requirements are specified as number of persons, necessary qualifications and skill characteristics. The degree of expertise is represented by skill levels. The qualifications and skills required to perform an activity are expressed by matrices $\mathbb{A} \times \mathbb{Q}$, $\mathbb{L} \times \mathbb{K}$:

$$RQ_{pr} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 3 & 1 \end{pmatrix} \qquad RK_3 = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}$$

For example, an activity has a qualification requirement $RQ_{pr}^{(3,2)} = 2$ and a skill requirement of $RK_3^{(1,2)} = 3$. The second column of RQ_{pr} represents the need for three persons with qualification Q_2 (column) for activity 3 (line). $RK_3^{(2,1)} = 3$ means that activity 3 requires at least 3 persons that have the skill 1 (column) of level 2 (line) or higher. In a feasible schedule the following conditions must hold:

$$\sum_{p \in AP} qr_{pq} ap_{ip} = RQ^{(i,q)}, \quad \forall \ p \in AP, (i,q) \ \in \mathbb{A} \times \mathbb{Q}$$

$$\sum_{p \in AP} SM_p^{(l,k)} ap_{ip} \geq RK_i^{(l,k)}, \quad \forall \ (l,k) \ \in \mathbb{L} \times \mathbb{K}$$

ap_{ip}: Binary variable *ap_{ip}* is "1" if a person $p \in \mathrm{AP}$ is assigned to the activity *i*, otherwise 0.

qr_{pq}: *Binary variable qr_{pq}* is "1" if a person $p \in \mathrm{AP}$ has the qualification *q*, otherwise 0.

The inclusion of persons and ressources leads to an extension of the Petri net to a Couloured Petri net. Thereby, different colours of tokens represent the qualification and skills of a person.

Based on the Petri-net models three classes of decision variables of a service oriented optimization problem can be derived: "*activity configuration*", "*decision configuration*" and "*iteration configuration*":

- *Activity configuration*: For each activity to be performed an activity configuration is defined. The configuration consists of:
    - Starting time t_i: The value defines the starting time of activity i in relation to the degree of completion of the predecessor activities j of i (activation time of transition d_{D_i}). Within a predetermined valid range a value z_i is randomly selected during the optimization process:
        - $z_i < 1$; Overlapping of activities i and j,
        - $z_i = 1$; Starting time of activity i is end time of activity j,
        - $z_i > 1$ Break between end of activity j and start of activity i.
    - *Duration of Activity* d_i: The parameter describes the amount of work necessary for a full implementation of activity i. The stochastic parameters (variance of estimation errors) and deterministic parameters (mode of execution, skill level of person) are considered.
    - *Number of Actors* AP_{An}: The parameter defines the number of person for the activity execution. The value will always range from a minimum to a maximum value.
    - *Actor-Constellation* AP_K: The value references to the characteristic of at least one actor who has to execute the activity. Thereby, the assigned actors fulfil the formal qualifications and the competence levels required.
    - *Importance* I_w: Importance of an activity which is communicated to the persons of the actor-constellation AP_K.
    - *Importance* I_s: Importance of the service process, all activities of a service process has the same value.
    - *Deadline* t_{dead}: The value of t_{dead} is used to calculate the urgency of an activity execution.
- *Decision configuration*: A decision configuration defines the result of an OR respectively XOR decision. Every OR and XOR relation of a service model posses a specific decision configuration. The configuration is equivalent to the result of competing transitions for firing in the corresponding Petri net model of a decision (cf. Fig. 5).
- *Iteration configuration*: The iteration configuration determines the occurrence of an iterative execution of one or more activities as well as the characteristic of the iteration loop. For each iteration node of a C3 model an individual value of the iteration configuration is defined (cf. Fig. 5).

The parameters of each class are specified and modified during an optimization process to identify the Pareto set $X^* \subseteq X$. Based on a complete set of decision variables and under consideration of the non-variable parameters of the service model (restrictions of the planning problem) a valid plan is developed. Thereby a specific set of configurations of a service model must always lead to the same plan characteristic to ensure reproducibility of plan quality.

5. Software System

Due to the genetic coding a transformation of all *activity*-, *decision*-, and *iteration configurations* (GP-Mapping) to a detailed service process is needed. Therefore, we developed five operations "*definition of configurations*" "*generation of flowchart"*, "*redesign flowchart*", "*developing service plan*" and "*multiobjective evaluation*" (cf. Fig. 6). The interaction of the operations describes the evolutionary process based on stochastic modification, decoding and evaluation. The components of the introduced general MOEA algorithm are small parts in these operations.

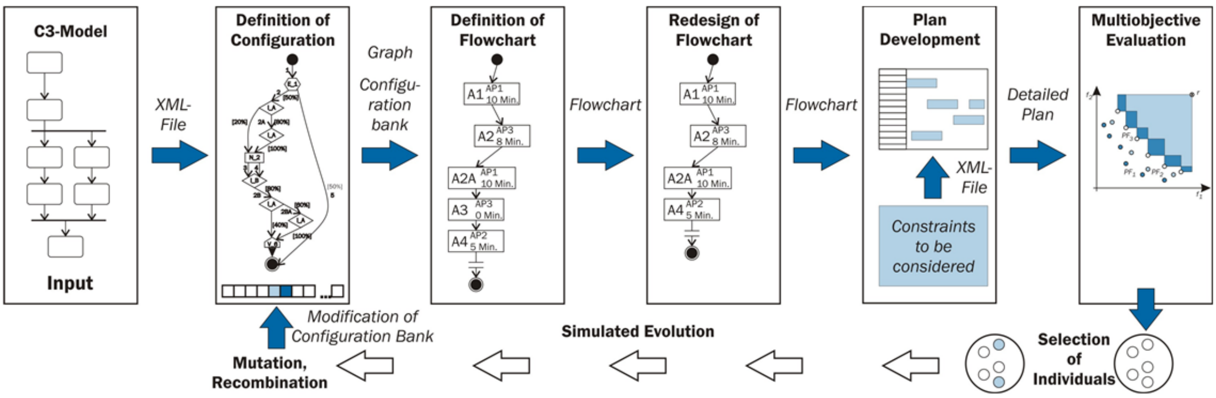

Fig. 6: Software Concept for a MOEA to optimize actor-oriented service modells

To introduce the main operations, "*definition of configurations*" "*generation of flowchart*", and "*developing service plan*" the semi-formal service process model in Fig. 7 is considered. The process structure is derived from a real-world service process.

5.1. Definition of configuration

The implemented module defines all activity, decision and iteration configurations based on the XML-file of the service process (semi-formal C3 model). Thereby, a Depth-First Search algorithm is used to develop a graph. The graph depicts all valid combinations of routing characteristics for the service process and describes all possible sequences of activities. We distinguish between a complete graph which includes all variants of decision and iterations and a sub-graph which describes only one valid service process characteristic. For the latter the decision about the occurrence of an iteration or the characteristic of a XOR/OR element in the process is already made. A graph consists of arcs, which represents activities, decision nodes *E*, iteration nodes *I*, synchronisation nodes *N* and branching elements *V.* Activities with a simple predecessor relation (one predecessor and one successor) are labeled on one arc. A complete graph for the service process is presented in Fig. 8 (left). It consists of a set of activity, decision and iteration configurations.

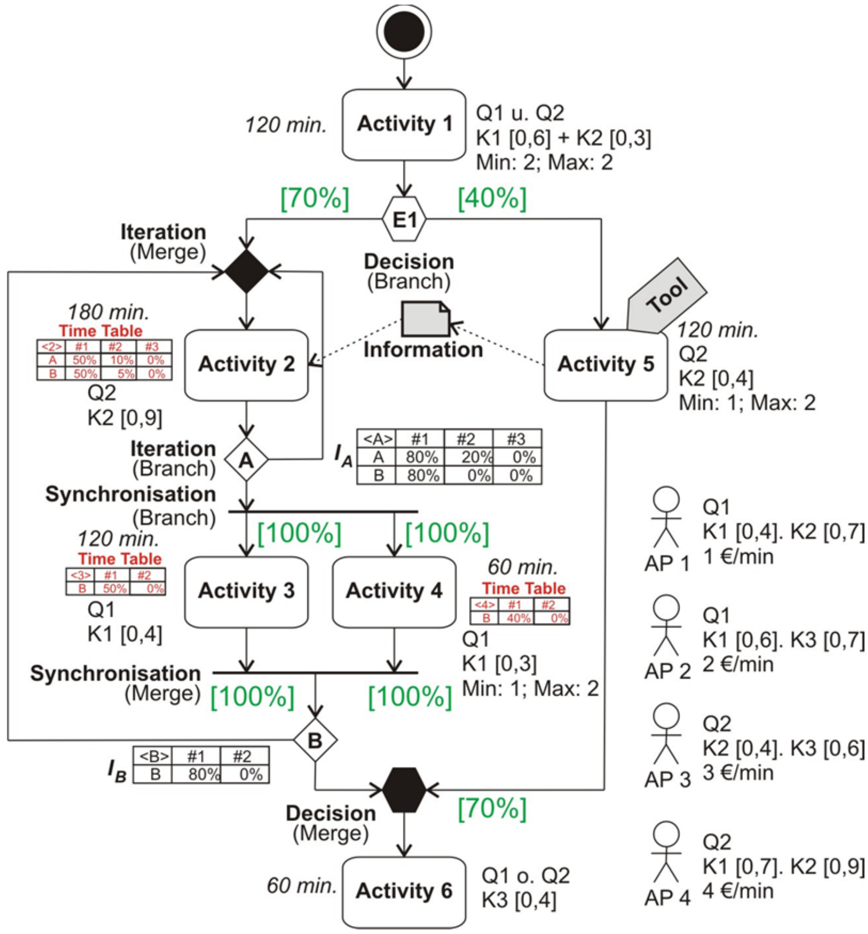

Fig. 7: C3 model of a service process

During the initiation phase of the MOEA all configurations of one graph are parameterized and a configuration bank is set up. A fully parameterized graph represents a valid individual. Therefore, the process is repeated until the pre-defined population size is reached. Thereby, the structure of the complete graph is identified at the beginning and afterwards only the configuration bank is modified. Storing information in a bank guarantees that a specific parameter of an activity *i* (e.g. duration of activity *i*) is at the identical bank position for each individual of the population. This characteristic is especially important, because mutation and recombination operators use the values of the bank to generate new solutions. With a fixed position it is guaranteed that only valid parameter characteristics are defined. Thereby, the configuration parameters are probability values representing a specific solution. Probability values are chosen to make recombination and mutation much easier.

5.2. Generation of flowchart

The module generates a flowchart of the corresponding service process originating from one specific set of configuration (individual) (cf. Fig. 8 right). The (abstract) schedule includes only those activities that are in accordance with the characteristic of the activity, decision and iteration configurations. Thus, only sequences and parameters of activities are displayed which are part of this specific service scenario, derived from the selected configuration. Furthermore, the final characteristics of the activities to be executed are defined based on the entries of the configuration bank.

Due to the interrelations between parameters (e.g. skill of person determines the expenditure of work) the values are calculated and assigned to each activity. It has to be taken into account, however, that the generated flow chart is not a detailed plan of a service process. For example, information about the absolute start point of an activity as well as the effects of the behavior of persons and their working time is missing.

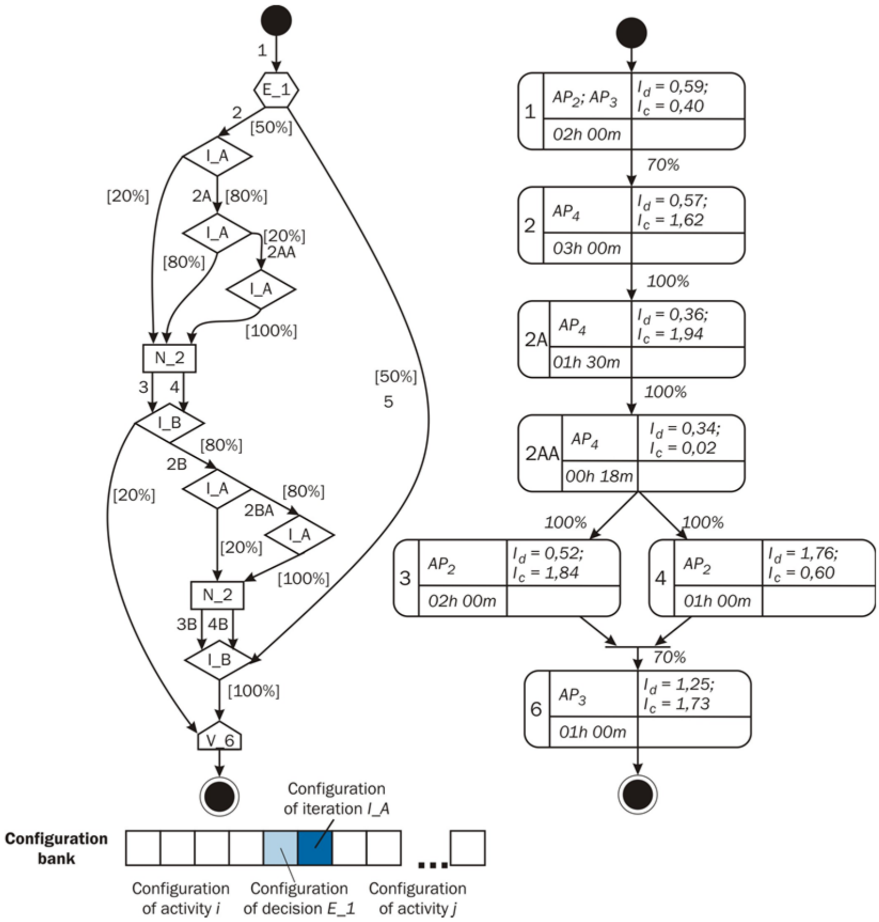

Fig. 8: Graph of service process structure (left) and flowchart (right)

5.3. Service plan development

The transformation of a flowchart into a detailed plan of a service process requires the identification of status changes within the process caused by persons. Therefore, the concept of an event queue was developed and implemented. The occurrence of an event always results in an organization of tasks (activities) by a person. At the time all predecessors of an activity are fulfilled, the task shows up in the task pool of the person listed in the activity configuration. As a result, a task pool of a person can contain a number of tasks with varying processing statuses due to interruption. If there is the choice between several tasks, the person organizes his or her task pool using the introduced priorization algorithm. The result determines the activity sequence. Three types of tasks are distinguished: 1) *Task unprocessed*: remaining effort = 100%; 2) *Task interrupted:* remaining effort < 100%) 3) *Task active*: The task is currently processed.

The organization of a task pool by a person is initiated by occurring events. We distinguish between various types of events:

- *New activity in pool*: The predecessors of an activity are sufficiently executed and the feasibility of the inserted activity is given.
- *Complete execution*: The execution of an activity leads to a remaining effort of 0.
- *Request for a cooperative activity execution:* Another person starts a request regarding a cooperative activity execution (defined in the activity configuration). Both persons negotiate a cooperative execution (passive negotiation). Only if both persons assign the highest priority to this specific task, the activity will be executed.
- *Start of working time*: At the beginning of a work day or after the daily break the person organizes the task pool.
- *Termination of cooperative activity*: If a cooperative activity is currently processed and one person stops the execution, all involved persons have to reorganize their task pools.
- *Period of time*: Without any other event, the person organizes the task pool after the expiration of a defined time.

The event based organization of a task pool leads to the continuation of the current activity or the initial start or restart of another activity. During a service provision several persons organize their task pools at different times. Due to the degree of complexity and uncertainty an activity can only be placed in the plan after a new event for a person occurred and the execution of the current activity is interrupted (cf. Fig. 9). Thus, the availability of persons and their behavior to organize the work determine the execution times of an activity and therefore the exact placement in the plan. We would like to underline that based on the decision variables of the MOEA the dynamics of the modeled service provision results of the simulated behavior of persons.

5.4. Plan evaluation and details of MOEA components

Our aim was to develop and to use a MOEA that fairly evaluates the non-dominated and dominated solutions with a limited number of points on each front. Furthermore, we wanted to diminish the parameterization of a fitness evaluation technique. Therefore we use the SMS-EMOA developed by Emmerich et al. (2005) which combines ideas of the NSGA-II (Deb et al., 2002) and archiving strategies (Zitzler et al., 2001). The SMS-EMOA is a steady-state evolutionary algorithm with constant population size that uses non-dominated sorting for ranking individuals and a hypervolume value for environmental selection (removing of the worst solution). A steady-state scheme seems to be well suited for a service based scheduling problem, since it can achieve a high diversity of the solutions on a front and it allows a graphical description of the fitness value (Emmerich et al. 2005). The basic algorithm of the SMS-EMOA is described in Emmerich et al. (2005). Starting with an initial population of μ individuals, every individual represents a valid set of *activity*, *decision* and *iteration configuration.* A new set of configuration for one individual is generated by SBX recombination and Polynominal mutation operators. To ensure the generation of valid solutions the genetic operators were modified for the scheduling problem. A new (modified) individual enters the population of the next generation *t*+1 if it dominates another solution or if it increases the hypervolume value covered by all individuals of the population.

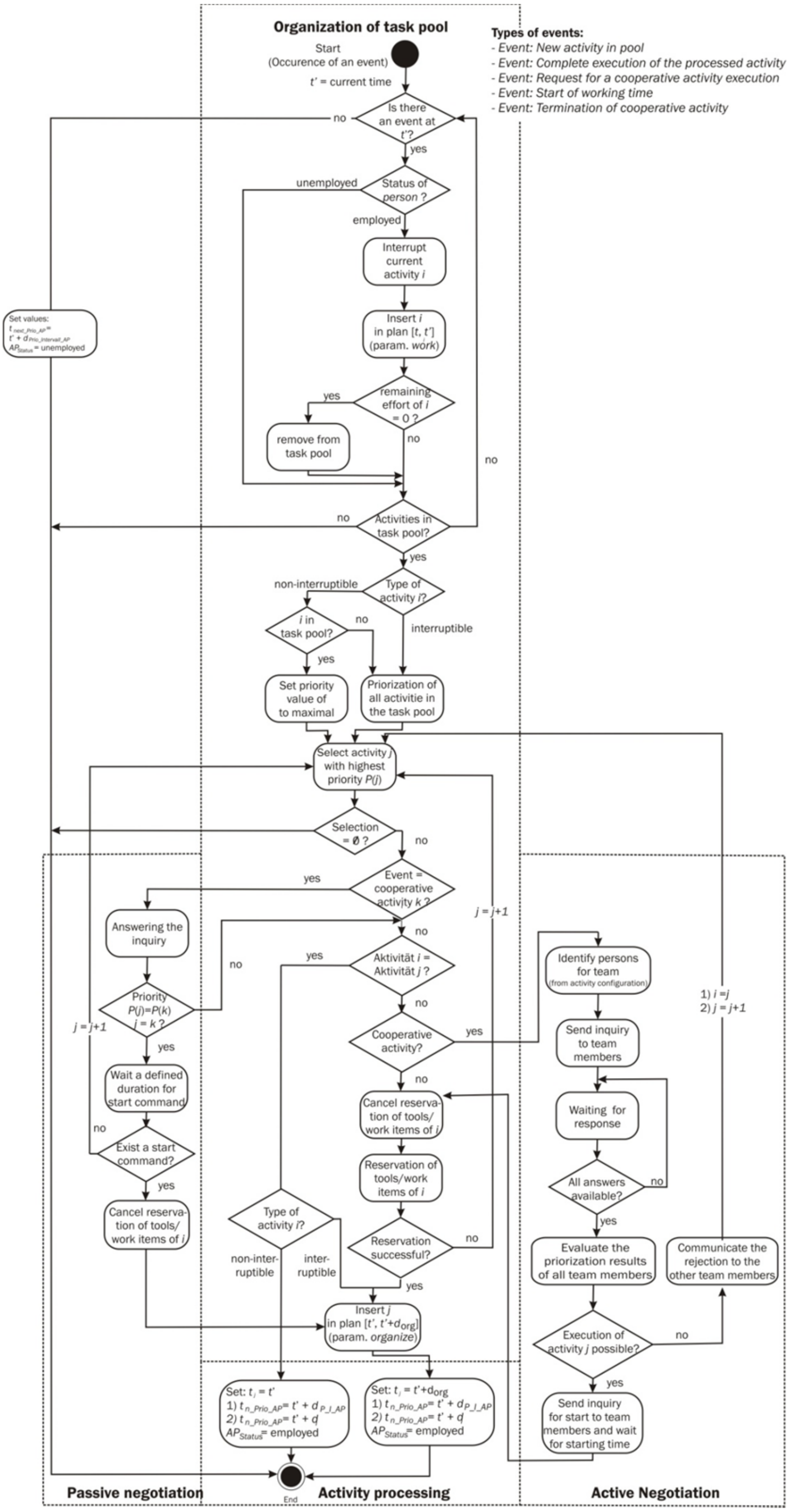

Fig. 9: Sub-Algorithm of the Planer – organization of the task pool of a person

The population of the implemented SMS-EMOA keeps dominated and non-dominated individuals without a separate archive. The population size is kept constant. Therefore, a rule-based selection algorithm is implemented to decide which individual is eliminated. First, the fronts of a solution for the population of size μ+1 are computed using the non-dominated sort algorithm (Deb et al., 2002). The first front contains all non-dominated individuals. Afterwards the individual on the last front $F_k = \{s_1, \dots, s_k\}$ (highest value) with the lowest hypervolume value Δs is deleted (Emmerich et al., 2005). For a two dimensional objective vector $y = (y_1, y_2)$, (e. g. service cost and service time) the individuals of the last front are sorted according to the first value y_1 and due to the non-dominated relations on a front the solutions are sorted in descending order concerning y_2.For the two dimensional case Δs is calculated as follows:

$$\Delta s(s, F_k) \coloneqq \big(f_1(s_{i+1}) - f_1(s_i)\big)\big(f_2(s_{i-1}) - f_2(s_i)\big)$$

6. Results of an optimization study

First, we investigated the progression of service cost and service-time over recent generations. The population size was set to 100 individuals. The analysis of the non-dominated solutions indicates a positive correlation between the number of generations and the achieved plan quality (cf. Fig. 10). Thereby, a front contains all non-dominated solutions of a population. It should be further noted that the global optimum for the fitness value does not appear until several generations. The stopping criterion was set to 2000 generations. Nevertheless, the fitness values are in line with the expected results, that activity sequence $A_1 \rightarrow A_5 \rightarrow A_6$ represents the global optimum (cf. Fig. 11).

The behavior of persons incorporated in the service model has an impact on the activity sequence. Fig. 12 shows the difference between a non- and a dominated solution due to different decision variables and a derived working strategy of the persons.

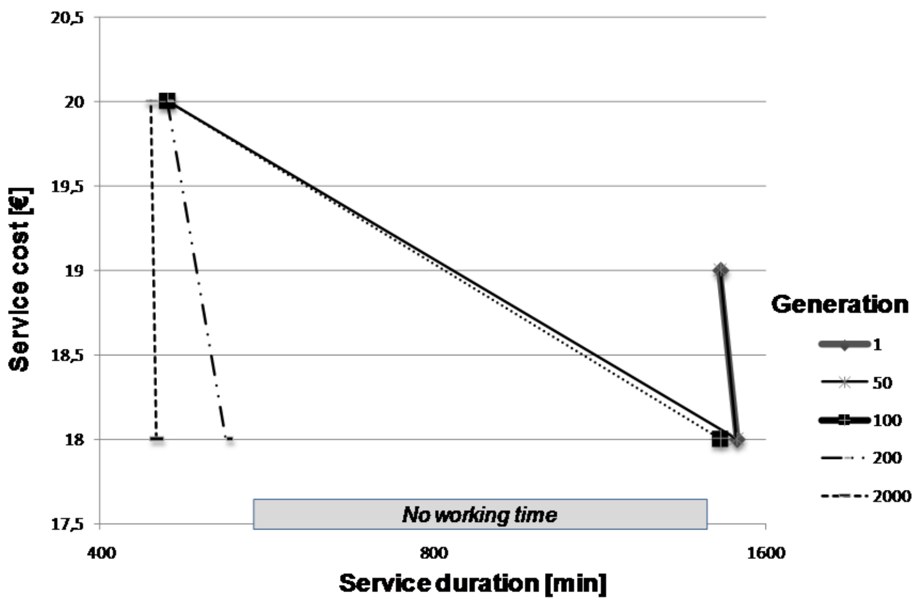

Fig. 10: Non-dominated solutions (Pareto-front) of different generations

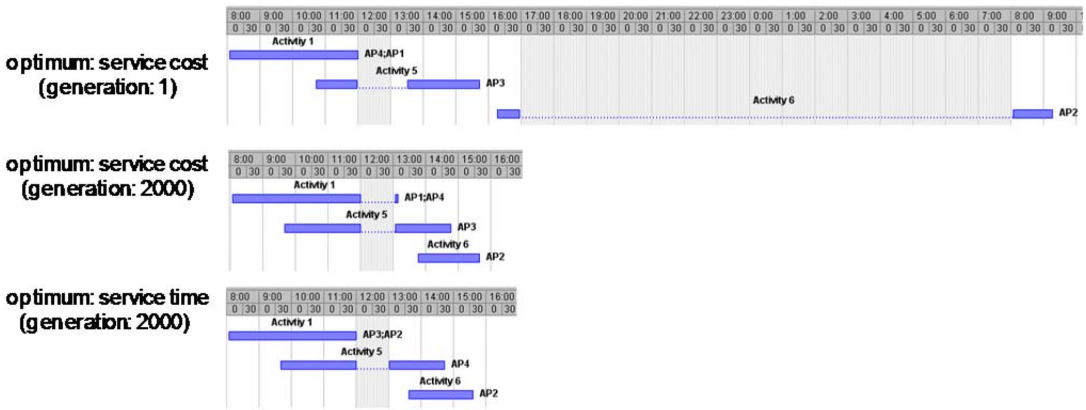

Fig. 11: Optimal plans of a service provision

A service process without iterative activity processing must be considered as a baseline schedule. In reality, some tasks may be forced to be repeated due to changes in the outcome of upstream tasks. In order to accommodate this problem, the characteristic of the iteration configurations of one individual is calculated from the probability distribution of each iteration loop. It is considered in the approach that the probability of a feedback loop can decrease or increase during multiple executions of iterations. Also the workload needed to iteratively execute an activity is varied. Two iteration configurations (I_A, I_B) are part of the service process model (Fig. 7). For the given parameterization the following activity sequences for A_2, A_3 and A_4 are valid: 1) $A_2 \rightarrow A_3 \& A_4$; 2) $A_2 \rightarrow A_{2A} \rightarrow A_3 \& A_4$; 3) $A_2 \rightarrow A_{2A} \rightarrow A_{2AA} \rightarrow A_3 \& A_4$; 4) $A_2 \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{3B} \& A_{4B}$; 4) $A_2 \rightarrow A_{2A} \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{3B} \& A_{4B}$; 5) $A_2 \rightarrow A_{2A} \rightarrow A_{2AA} \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{3B} \& A_{4B}$; 6) $A_2 \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{2BA} \rightarrow A_{3B} \& A_{4B}$; 7) $A_2 \rightarrow A_{2A} \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{2BA} \rightarrow A_{3B} \& A_{4B}$ 8) $A_2 \rightarrow A_{2A} \rightarrow A_{2AA} \rightarrow A_3 \& A_4 \rightarrow A_{2B} \rightarrow A_{2BA} \rightarrow A_{3B} \& A_{4B}$.

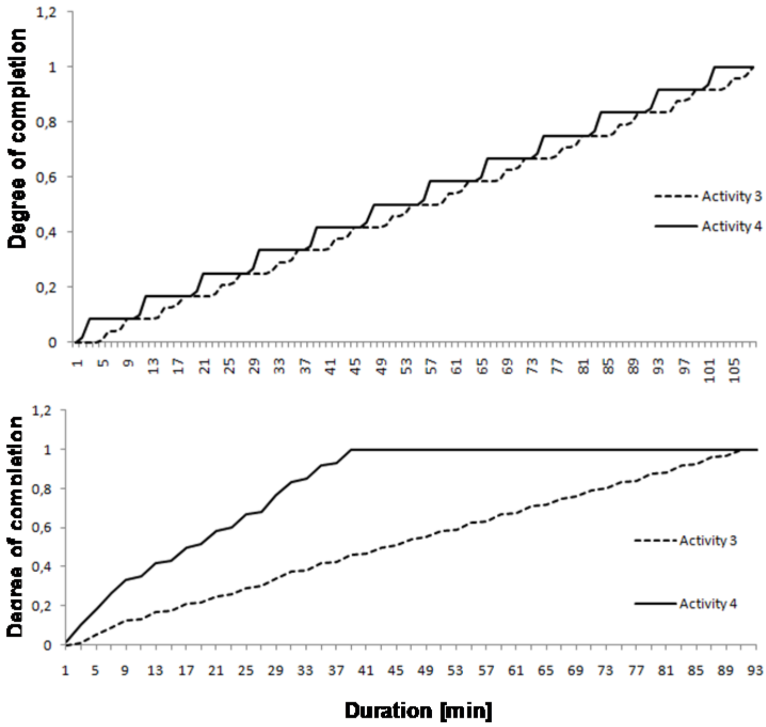

Fig. 12: Effects of bounded decision behavior on the processing of activities

In our view, the fitness values (quality of a plan) of different service scenarios with stochastic iterations are incomparable. To overcome the problem that solutions with iterations are deleted from the population due to poor values for the objectives "*service cost*" and "*service time*" we use the island concept of parallel evolutionary multi-objective optimization (pMOEA) (Coello Coello et al., 2007). Thereby, every island contains all individuals with a specific iteration characteristic. The relative quantity of individuals on an island compared to the population is determined by the likelihood of their occurrence of this specific scenario. In accordance with the pMOEA concept an autarkic evolutionary process can be set up on each island. The same or different parameterized MOEA algorithms try to identify the non-dominated individuals on each island. This ensures, that our approach guarantees the parallel optimization of all valid service scenarios. By meeting the stopping criteria, the non-dominated plans for each scenario are presented and the service manager can choose a plan based on his or her risk preference. The non-dominated solutions (Pareto-front) of each island after 2000 generations are presented in Fig. 13.

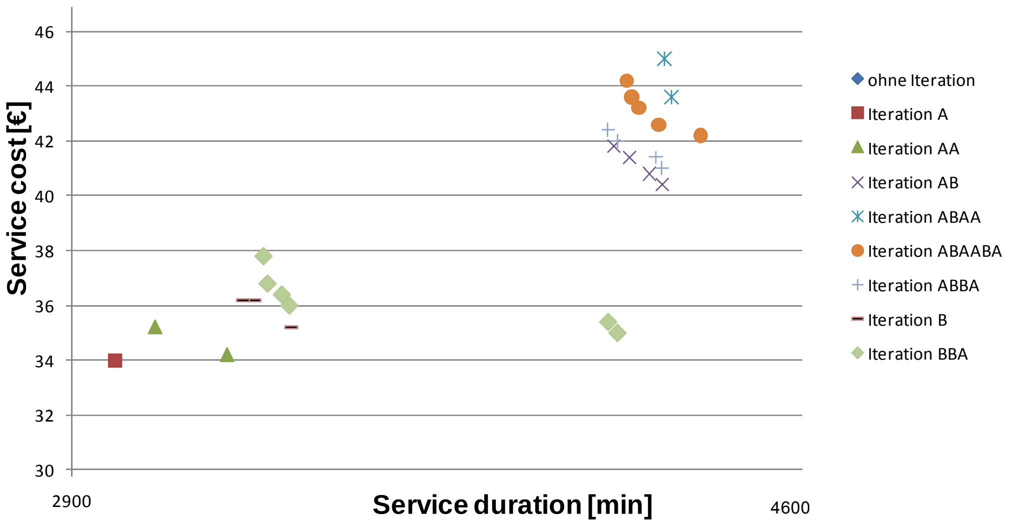

Fig. 13: Non-dominated individuals (Pareto-fronts) of valid iteration characteristics

7. Conclusion and Future Work

In the previous sections, a MOEA was introduced to solve the Resource Constraint Service Scheduling Problem. Determining the sequence of activities with given and indefinite predecessor constraints as well as uncertainty regarding the time-on-task is a NP-hard scheduling problem. The objective was to minimize the service lead-time and costs through the improvement of activity sequences, assignment of persons, under certain constraints (availability; qualification, skills, decision behavior of actors etc.). The developed methods enabled us to identify and evaluate optimal service processes under uncertainty. The algorithm was heavily influenced by the findings of Deb et al. (2002) Emmerich et al. (2005) and Zitzler et al. (2004) in the multiobjective optimization domain. The introduced MOEA offers a novel concept that is able to encode random performance fluctuations and unpredictable behavior of persons as a genetic code. Finally, a validation study was carried out for a small service project. Due to an existing documentation about the service process, detailed information about the task processing and the amount of work was able to be acquired. It was shown that the approach offers managers of service organizations a suitable technique for the quantitative comparison of alternative service scenarios at an early planning stage. An additional comparison of computed service processes with those created by humans demonstrated moreover that it is difficult for managers to evaluate all valid solutions of a complex service planning problem (Tackenberg et al., 2010). In future papers we will present more details about the planer (GP mapping). Furthermore, additional optimization studies of more complex service processes with a larger number of coupled activities and more persons will be carried out. Thereby, the influence of different skill levels on the duration of a service activity will be investigated in empirical studies to validate the model. In addition, further key figures of service productivity will be identified and integrated into the service model.

Acknowledgment

The research project “WiDiPro” is funded by the German Federal Ministry of Education and Research according to Grant No. 01FL10011.

References

Coello Coello, C. A.; Lamont, G. B.; van Veldhuisen, D. A. (2007): Evolutionary algorithms for solving multi-objective problems. 2nd. New York: Springer.

Dahal, K. P.; Cowling, P.; Tan, K. C. (2007): Evolutionary Scheduling. Berlin, Heidelberg: Springer-Verlag.

Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002): A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6 (2), pp. 182-197.

Emmerich, M.; Beume, N.; Naujoks, B. (2005): An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In: Coello Coello, C. A.; Aguirre, A. H.; Zitzler E. (ed.): Evolutionary multi-criterion optimization. Third international conference, EMO 2005. Berlin: Springer, pp. 62-76.

Firat, M.; Hurkens, C. (2010): A combinatorial approach to multi-skill workforce scheduling. Working Paper: Eindhoven University of Technology.

Funke, J.; Glodowski, A.-S. (1990): Planen und Problemlösen: Überlegungen zur neuropsychologischen Diagnostik von Basiskompetenzen beim Planen. Zeitschrift für Neuropsychologie 2, pp. 139-148.

Girault, C.; Valk, R. (2003): Petri nets for systems engineering: a guide to modeling, verification, and applications. Berlin: Springer.

Grönroos, C.; Ojasalo, K. (2004): Service productivity: Towards a conceptualization of the transformation of inputs into economic results in services. Journal of Business Research 57 (4), pp. 414-423.

Hayes-Roth, B.; Hayes-Roth, F. (1979): A Cognitive Model of Planning. Cognitive Science 3 (4), pp. 275-310.

Killich, S.; Luczak, H.; Schlick, C. M..; Weissenbach, M.; Wiedenmaier, S.; Ziegler, J. (1999): Task modelling for cooperative work. Behaviour & Information Technology 18 (5), pp. 325-338.

Lasshof, B. (2006): Produktivität von Dienstleistungen. Mitwirkung und Einfluss des Kunden. Wiesbaden: Deutscher Universitäts-Verlag.

Miller, G.A.; Galanter, E.; Pribram, K.H. (1960): Plans and the structure of behavior. New York: Holt.

Murata, T.; Ishibuchi, H.; Tanaka, H. (1996): Genetic Algorithms for Flowshop Scheduling Problems. Computers & Industrial Engineering 30 (4), pp. 1061-1071.

Steel, P.; Koenig, C. J. (2006): Integrating Theories of Motivation. The Academy of Management Review 31 (4), pp. 889-13.

Tackenberg, S.; Rieder, S.; Duckwitz, S.; Schlick, C. M. (2010): Evaluation of a Genetic Algorithm based Service Productivity Optimization. In: Ershi, Qi (ed.): 2010 IEEE 17th International Conference on Industrial Engineering and Engineering Management. Vol. 2. Piscataway, NJ: IEEE, pp. 1528-1533.

Tan, K. C.; Khor, E. F.; Lee, T. H. (2005): Multiobjective evolutionary algorithms and applications. London: Springer.

Winkelmann, K.; Luczak, H. (2006): Modelling, Simulation and Prospective Analysis of Cooperative Provision of Industrial Services Using Coloured Petri Nets. International Journal of Simulation 7 (7), pp. 10-26.

Zitzler, E.; Laumanns, M.; Bleuler, S. (2004): A Tutorial on Evolutionary Multiobjective Optimization. In: Gandibleux, X.; Sevaux, M.; Sörensen, K. (ed.): Metaheuristics for multiobjective optimisation. Berlin: Springer, pp. 3-38.

Zitzler, E.; Laumanns, M.; Thiele, L. (2001): SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Working Paper: ETH: Zürich.

Authors

Dipl.-Wirt.-Ing. Sven Tackenberg, Dipl.-Inf. Peter Schmitz,
Dipl.-Ing. Dipl.-Wirt.Ing. Sönke Duckwitz,
Univ.-Prof. Dr.-Ing. Dipl.-Wirt.-Ing. Christopher M. Schlick
Institute of Industrial Engineering and Ergonomics, RWTH Aachen University
Bergdriesch 27, 52062 Aachen, Germany
{s.tackenberg},{p.schmitz},{s.duckwitz},{c.schlick}@iaw.rwth-aachen.de