A New Genetic Algorithm to Solve Effectively Highly Multi-Objective Problems: POGA

Francesco di Pierro, Soon-Thiam Khu, Slobodan Djordjević and Dragan A. Savić

F.di-Pierro@exeter.ac.uk
S-T.Khu@exeter.ac.uk
D.Savic@exeter.ac.uk
S.Djordjevic@exeter.ac.uk

Report Number: 2004/02

July, 2004

Centre for Water Systems
University of Exeter
North Park Road
Exeter, EX4 4QF
Devon
United Kingdom.

This work was funded by the by European Union's Framework 5 Programme, TiGrESS project, EVG3-2001-00024

INTRODUCTION

Since the pioneering work of Holland [14], Genetic Algorithms (GA) have become one of the most popular optimization techniques. The main reasons behind this success can be attributed to their effectiveness in exploring massive search spaces with little tendency to be deceptively attracted by local optima, and inherent capability of handling both real and discrete variables. Furthermore, they are relatively easy to code and are somewhat problem domain independent, even though prior knowledge and fine tuning are often required to achieve best performances. Broadly speaking, GA is a search procedure inspired by natural selection and genetics and it is based on the concept of "survival of the fittest". Given an initial population of "individuals", recombination and selection are repeatedly performed until a set of *good enough* solutions is found.

The great potential of GA was indubitable and apparent to the scientific community, but it was only with Goldberg [13] that it became evident that they could be applied to a broad range of engineering problems. With Goldberg's work, the transition between single-objective GA to multi-objective GA (MOGA) was first paved and a manifold of methodologies have been flourishing in the literature ever since.

Many real-world optimization problems involve multiple objectives that need be considered simultaneously. If these objectives are conflicting, as it is usually the case, a single solution optimal for all the objectives can not be found. As a consequence, the assessment of the quality (fitness) of a set of solutions poses a serious problem and calls for a method that provides a formal definition of the qualitative notion of compromise. The great majority of MOGAs presented to date [5;10;15;16;20;22;23], solve this predicament through the concept of Pareto optimality (in the remainder of the paper the term Pareto optimality will be used quite interchangeably with Pareto efficiency), which is exploited to rank the solutions generated and therefore to drive the search for better ones.

It is well known that as the number of objectives of a Multi Objective Problem (MOP) increases, the number of Pareto efficient, i.e. equally good solutions, quickly becomes vast. As a consequence, all Pareto-based ranking techniques suffer the same limitations: the genetic algorithm search worsens (low selective pressure) and decision makers are presented with an overwhelming number of equally optimal solutions (parameter sets). While, to the best knowledge of the authors, the literature lacks a rather theoretical and unified approach to tackle the former issue, the latter has been investigated in depth. The main principle is to exploit the vague idea that the user has about what kind of solutions might be preferred to guide and focus the search. Coello Coello [3] and Branke [1] provide a rather comprehensive survey of attempts to handle preferences: following the definition proposed by Van Veldhuizen [21] who suggests the articulation of preferences might be *a priori*, *progressive* or *a posteriori* (i.e. performed before, during or after the optimization process) most Evolutionary Multi-Objective Optimization (EMO) approaches belong to the latter category. Although, it has been found that such methods, besides allowing for a more fine-grained selection of those alternatives that are considered the most interesting from the user's point of view, might also speed up the search [1;2], they all have been shown very sensitive to different problem domains [8] and to little variations of parameters that control the preferences specification [6].

Almost contemporarily, Das [7] introduces Preference Ordering (PO), a Pareto-based optimality condition. In effect, he shows that Preference Ordering is a generalization of Pareto efficiency and that it provides a more stringent criterion than Pareto dominance to compare solutions to MOPs.

Preference Ordering has been applied successfully in a number of studies as a post processing routine: once an optimization algorithm has found a set of Pareto optimal solutions, PO can be applied to sieve through them so that the decision maker is presented with a manageable number of high quality solutions to assess. Nonetheless, despite its interesting features, to the best knowledge of the authors PO has never been exploited as an optimality condition to drive the search of an MOGA.

This paper presents a new MOGA, the Preference Ordering Genetic algorithm: POGA. It is largely inspired by NSGA-II [10], a most acknowledged MOGA that has proved successful in a number of applications, but in contrast it embodies Preference Ordering in its ranking scheme to drive the search for better solutions. The paper is organized as follows: the first section provides the theoretical background and a detailed description of the algorithm is given in the second one. The third and fourth sections compare the results of the application of the two algorithms to the optimization of a test function and to the automatic calibration of an urban drainage model respectively. A particular emphasis is given to the issue of scalability and to support that POGA is particularly efficient to tackle highly dimensional MOPs. Finally, the results are discussed and unexplored research issues suggested.

FROM PARETO EFFICIENCY TO PREFERENCE ORDERING

The multi-objective optimization problem (MOP) can generally enough be stated as [4;7]:

Definition (MOP). Find a vector $\boldsymbol{x}^* = [x_1^*, x_2^*, ..., x_n^*]^{\mathrm{T}}$ such that

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in D} \boldsymbol{f}(\boldsymbol{x}) = \min_{\boldsymbol{x} \in D} [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]^{\mathrm{T}} \\ \boldsymbol{g}_i(\boldsymbol{x}) \geq 0 \quad i = 1, 2, ..., l \\ \boldsymbol{h}_i(\boldsymbol{x}) = 0 \quad i = 1, 2, ..., L \end{aligned} \tag{1}$$

where n is the dimension of the decision variables vector, m is the number of objectives for the MOP, l is the number of inequality constraints $\boldsymbol{g}(\boldsymbol{x})$, L the number of equality constraints $\boldsymbol{h}(\boldsymbol{x})$. D represents the *search space* (also referred to as *decision space*), i.e. the Euclidean space that is defined by the set of all n-tuples of real numbers, denoted by $\mathbb{R}^n$. This definition could be easily extended to situations where the variables take values on other sets. The constraints $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ act on D as to defining a subset Ω, which represent the *feasible region* for the MOP. Any point $\boldsymbol{x}$ in Ω represents a feasible solution; the function $\boldsymbol{f}$ maps Ω into the $\mathcal{F}$ that is a subset of the *objective space*, denoted by $\mathbb{R}^m$.

Equations (1) are general enough to accommodate different problem definitions (maximization instead of minimization and different forms of equality/inequality constraints) since it is always possible to rewrite the problem equations in the form introduced above.

Definition (Ideal solution). A point ${}^U\boldsymbol{x}^* = [{}^Ux_1^*, {}^Ux_2^*, ..., {}^Ux_n^*]^{\mathrm{T}}$ such that

$$\boldsymbol{f}({}^U\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in D} \boldsymbol{f}({}^U\boldsymbol{x}) = [\min_{\boldsymbol{x} \in D} f_1({}^U\boldsymbol{x}), \min_{\boldsymbol{x} \in D} f_2({}^U\boldsymbol{x}), ..., \min_{\boldsymbol{x} \in D} f_m({}^U\boldsymbol{x})]^{\mathrm{T}} \tag{2}$$

is the ideal solution and $\boldsymbol{f}({}^U\boldsymbol{x}^*)$ is the ideal vector (also referred to as utopia point or shadow minimum) for the MOP. In words, a point ${}^U\boldsymbol{x}^*$ is an ideal solution for an MOP, and consequently $\boldsymbol{f}({}^U\boldsymbol{x}^*)$ is the utopia point, if all the individual objectives are contemporarily optimized.

The problem defined by equations (1) is well posed only if the objective functions $f_1, \ldots, f_n$ are non conflicting. When this holds, if the ideal solution lays in the feasible region, i.e., if ${}^U\boldsymbol{x}^* \in \Omega$, then it is the solution to the optimization problem. Unfortunately, this is seldom the case. Most engineering

problems naturally lead to the concurrent optimization of a pool of conflicting objectives. In such situations, a single (utopical) solution is not attainable; instead, one might be interested in finding a set of good compromises ("trade-offs"). Pareto optimality provides with a formal definition of the intuitive yet qualitative notion of compromise.

Definition (Pareto Dominance and Pareto Optimality). The vector $\boldsymbol{f}(\tilde{\boldsymbol{x}})$ is said to dominate the vector $\boldsymbol{f}(\hat{\boldsymbol{x}})$, denoted $\boldsymbol{f}(\tilde{\boldsymbol{x}}) \prec \boldsymbol{f}(\hat{\boldsymbol{x}})$, if $f_i(\tilde{\boldsymbol{x}}) \leq f_i(\hat{\boldsymbol{x}})$ for all $i \in \{1,2,\ldots,n\}$ and $f_j(\tilde{\boldsymbol{x}}) < f_j(\hat{\boldsymbol{x}})$ for some $j \in \{1,2,\ldots,n\}$. A point $\boldsymbol{x}^*$ is said to be Pareto optimal or Pareto efficient for the MOP if and only if there does not exist $\boldsymbol{x} \in \Omega$ such that $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$. In words, a point $\boldsymbol{x}^*$ is Pareto efficient if there does not exist a point $\boldsymbol{x}$ in Ω that would achieve a better value for one of the objectives without worsening at least another.

It is easy to show that Pareto dominance induces a partition of the objective space. Without loss of generality we restrict our reasoning to the feasibility region Ω. Suppose, for instance, that within Ω there exists a point A; its coordinates in Ω define the regions where points that dominate A and that are dominated by A could lie. We denote these regions Ω_A^+ and Ω_A^- respectively. Figure 1 is a pictorial representation of Ω_A^+ and Ω_A^- for two and three dimensions.

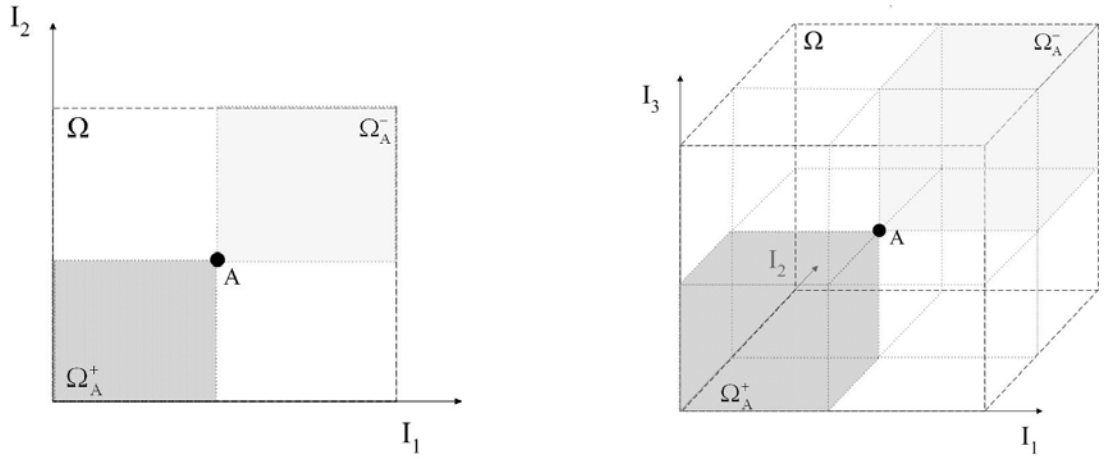

Figure 1: Representation of the partitioned induced by Pareto dominance onto the two and three-dimension objective spaces $\{I_1, I_2\}$ and $\{I_1, I_2, I_3\}$.

Definition (Pareto optimal set). For a given MOP, the Pareto optimal set denoted $\mathcal{P}^*$, is defined as:

$$\mathcal{P}^* := \{\boldsymbol{x} \in \Omega \mid \neg\exists\, \boldsymbol{x'} \in \Omega\ \boldsymbol{f}(\boldsymbol{x'}) \prec \boldsymbol{f}(\boldsymbol{x})\}$$

Definition (Pareto front). For a given MOP and a given Pareto optimal set $\mathcal{P}^*$, the Pareto front is defined as:

$$\mathcal{PF}^* := \{\boldsymbol{f} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})) \mid \boldsymbol{x} \in \mathcal{P}^*\}$$

As the number of objectives increases, the $\mathcal{PF}^*$ of an MOP quickly becomes vast. An intuitive explanation is suggested by Figure 1: moving from a two to a three-dimension objective space, $(\Omega_A^- + \Omega_A^-)$ grows at a lower rate than Ω.

This phenomenon has a great impact on the performance of the optimization algorithms that relies on Pareto optimality principle to drive their search for good solutions. A criterion more stringent than Pareto optimality, yet general, would be highly desirable. On these premise, Preference Ordering [7] comes to rescue.

Definition (Efficiency of order). A point $\boldsymbol{x}^* \in \Omega$ is considered efficient of order k if $\boldsymbol{f}(\boldsymbol{x}^*)$ is not dominated by any member of $\mathcal{PF}^*$ for any of the k-element subsets of the objectives. In words, a point is efficient of order k if is Pareto optimal in all the ${}^{m}\boldsymbol{C}_{k}$ subspaces of Ω obtained considering only k objectives at a time. It is clear that the efficiency of order m for an MOP with exactly m objectives simply enforces the Pareto optimality; therefore Preference Ordering is an extension of it.

Figure 2 illustrates the efficiency of order for a set of points in a three-dimensional objective space Ω: amidst this set, the point efficient of order 2 is the only point Pareto optimal for all the 3 two-dimensional objective spaces obtained by considering all combinations of the three objectives of Ω, taken two at a time.

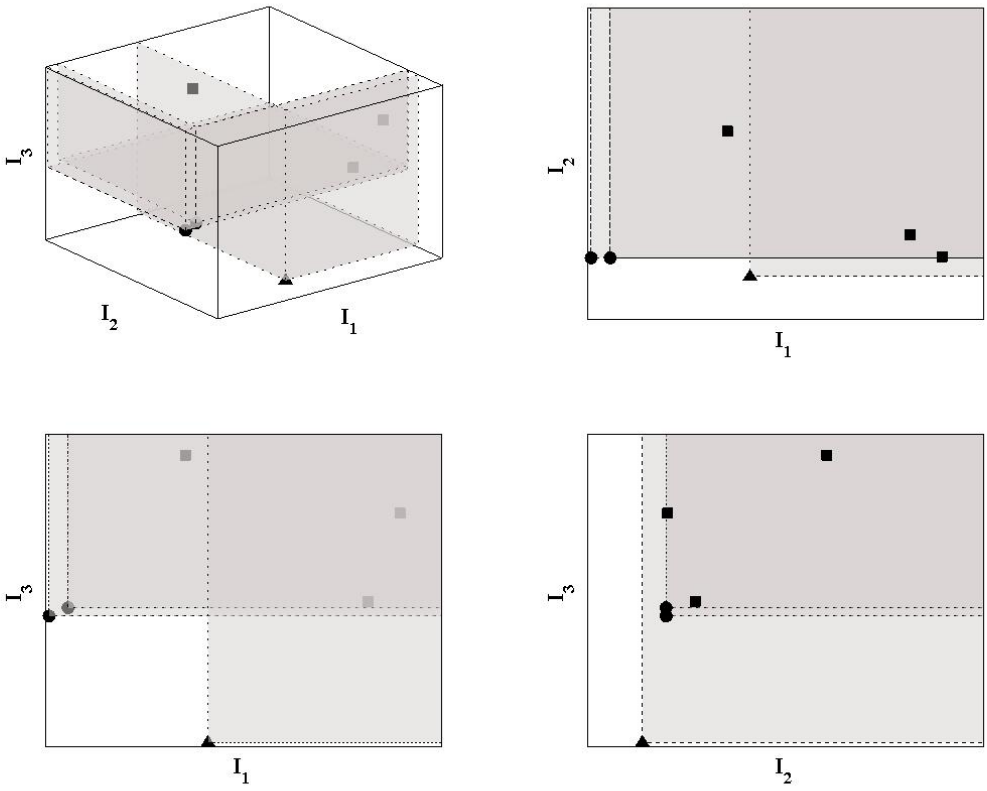

Figure 2: Pictorial representation of a set of points in a three-dimensional objective space and in its three two-dimensional projections onto the main planes. Squares and circles represent dominated and non dominated points respectively; the triangle represents the Pareto optimal point efficient of order 2.

Claim (*C1*): In a three-dimensional space there can be no more than one point efficient of order 2. This is a rather counterintuitive and interesting feature of Preference Ordering.

Proof: Suppose that A and B are two points in a three-dimensional objective space. Let us suppose that A and B are both efficient of order 2 and that $f_1(\mathrm{A}) < f_1(\mathrm{B})$. It follows that for the objective space $\{\mathrm{I}_1,\mathrm{I}_2\}$, the following relations must hold

$$f_1(\mathrm{A}) < f_1(\mathrm{B}) \ \wedge f_2(\mathrm{B}) < f_2(\mathrm{A}) \tag{3}$$

From the first relation of (3), it follows that for the objective space $\{I_1,I_3\}$ the following must hold

$$f_1(A) < f_1(B) \wedge f_3(B) < f_3(A) \quad (4)$$

and from the second relation (3) follows that for $\{I_2,I_3\}$ it must be

$$f_2(B) < f_2(A) \wedge f_3(A) < f_3(B) \quad (5)$$

The second relations of (4) and (5) cannot both be contemporarily satisfied, unless $f_3(A) = f_3(B)$. But according to relations (4) and (5), this implies that neither A nor B are efficient of order 2. In a similar way, it can be verified that the proof holds for $f_1(A) > f_1(B)$.

A graphical explanation of *C1* is presented in Figure 3. As introduced above, given a point A in an objective space Ω, Pareto optimality delimits Ω_A^+ and Ω_A^-. We denote ${}^{\{I_i,I_j\}}\Omega$, ${}^{\{I_i,I_j\}}\Omega_A^+$ and ${}^{\{I_i,I_j\}}\Omega_A^-$ their projections onto the two-dimensional spaces for every $i,j = 1,2,3$ with $i \neq j$, respectively. Let us assume that A is efficient of order 2. For any point B to be efficient of order 2, its projections onto the plane I_iI_j should lye in $({}^{\{I_i,I_j\}}\Omega - {}^{\{I_i,I_j\}}\Omega_A^+ - {}^{\{I_i,I_j\}}\Omega_A^-)$ for every $i,j = 1,2,3$ with $i \neq j$. It can be seen that A is the only such a point.

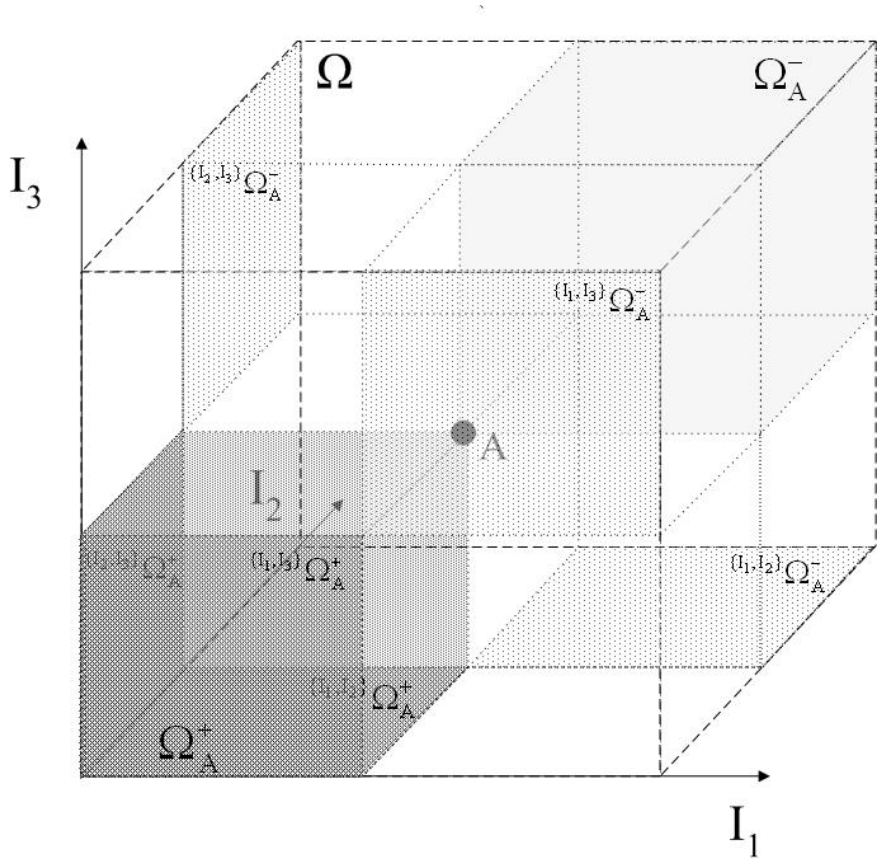

Figure 3: Graphical interpretation of Claim *C1*.

The condition of efficiency of order can thus be used to help reduce the number of points in a set by retaining only those that are regarded as "best compromises". In fact, it is intuitive to understand that the less extreme components a point has, the more likely it is to be efficient of order. When the number of points selected is still considerable, one might resort to an even more stringent criterion to sieve through:

Definition (Efficiency of order k with degree z): A point $\boldsymbol{x}^*$ is said to be efficient of order k with degree z if not dominated by any member of $\mathcal{P}^*$ for exactly z out of the possible ${}^{m}\boldsymbol{C}_{k}$ k-element subsets.

It is now worth pointing out that, as opposed to the condition of efficiency of order, the condition of efficiency with degree favours solutions that have extreme components. Therefore one should carefully orchestrate the cooperative usage of these conditions of efficiency. Let us suppose that the MOP at stake has m objectives. If the Pareto set $\mathcal{P}^*$ identified is too numerous, one could resort to the condition of efficiency of order to identify the subset $\mathcal{P}^*_k$ that consists of solutions in $\mathcal{P}^*$ that are efficient of order k. If $\mathcal{P}^*_k$ is still too numerous, one could retain only those solutions that are efficient with the highest degree, i.e. solutions in $\mathcal{P}^*_k$ that are not dominated by any point in $\mathcal{P}^*$ for the most of the possible ${}^{m}\boldsymbol{C}_{k-1}$ (k-1)-element subsets. If $k = m$, i.e. if there are no solutions in $\mathcal{P}^*$ that are efficient of order m-1, applying the condition of efficiency with degree results in the identification of the solutions that lay at the edge of the Pareto front. Therefore, one should be cautious in considering which option is most suitable for a particular purpose.

The Multi Objective Genetic Algorithm introduced in this paper, POGA, embodies Preference Ordering in the form of the condition of efficiency of order into its ranking scheme. The following section describes in details the structure of the algorithm.

THE PREFERENCE ORDERING GENETIC ALGORITHM: POGA

As stated in the introduction, the motives behind the development of a new technique for Multi Objective optimization lye on the inadequacy of Pareto-based GA to deal with highly dimensional objective spaces. In these situations in fact, Pareto-dominance ranking approaches fail in maintaining a sufficiently high selective pressure throughout the search process, which consequently stalls prematurely. On these premise, we present the Preference Ordering Genetic Algorithm: POGA.

POGA inherits from NSGA-II its evolutionary traits: selection, recombination and composition are retained unaltered and they are performed in the same order: Figure 4 presents the flowchart of POGA, which can be summarized as follows:

(i) Generate a random initial population P_0 of size s;
(ii) Evaluate the population (objectives and constraints) using a fast sorting method (fast-non-dominated-sort);
(iii) Rank the individuals of P_0;
(iv) Create a child population through selection and recombination operators;
(v) Combine the child and parent population into a compound population;
(vi) Rank the compound population;
(vii) From the compound population, place the s best-ranked solutions to the next generation population;
(viii) If there is a tie between which equally-ranked solutions to be taken from the compound population to fill the next generation population, resort to the crowding distance. This step favors solutions that rest in less crowded portions of the objective space so as to ensure a well spread Pareto front;
(ix) Repeat steps (v) to (viii) until the stopping criteria are met.

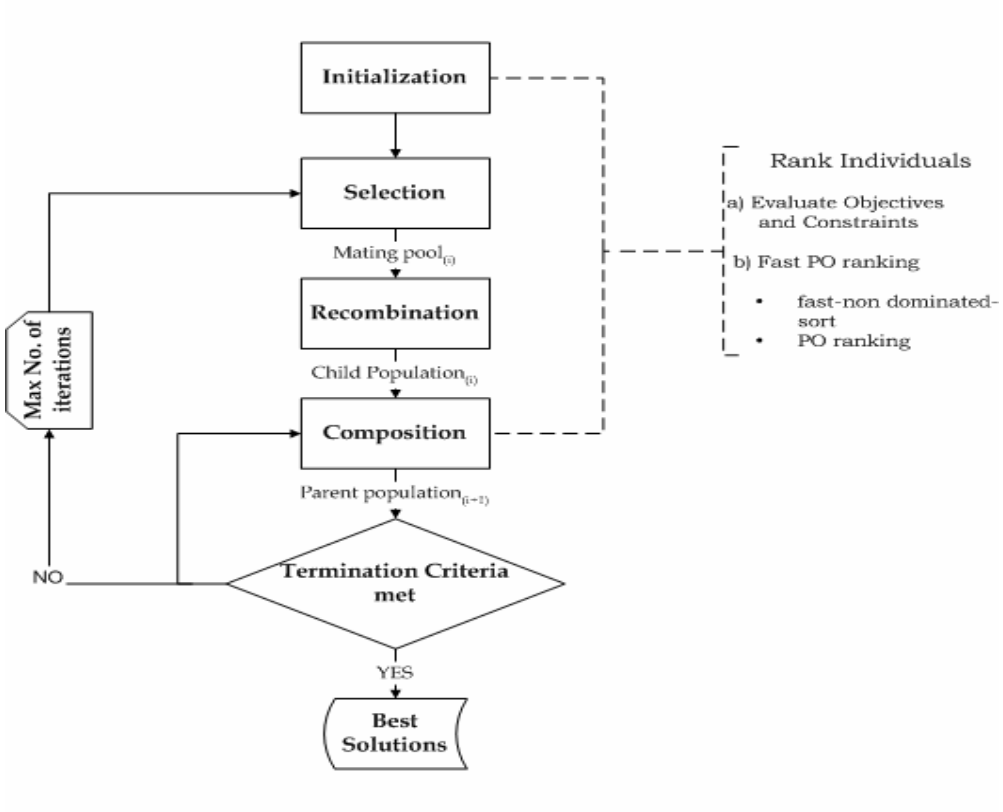

Figure 4: Flowchart of POGA.

The two algorithms differ fundamentally in the ranking procedure adopted. While NSGA-II exploits the Pareto efficiency criterion to order the individuals of a population (steps (iii) and (v)), POGA relies on Preference Ordering in the form of condition of efficiency of order. In the remainder of the paper we denote this condition PO_k to distinguish it from that of order and degree, which is denoted $PO_{k,z}$. When such a distinction is not relevant for the reasoning, PO is employed instead. Figure 5 exemplifies the ranking procedure performed by POGA. At the iteration t, the parent population P_t and the child population ${}^{C}P_t$ are combined into the compound population C_t. First, Pareto efficient individuals of C_t are given rank 1 and grouped into the subset R_1. Pareto efficiency (PE) is then applied to $\{C_t - R_1\}$ to identify R_2; the process continues until all individuals C_t have been ranked, i.e. the subset $\{C_t - R_r\}$ is empty, where r is the last (worst) rank computed. Last, the subset of the best individuals, R_1, is further ranked through Preference Ordering. The pseudo code of the ranking procedure implemented in POGA follows:

```
for every generation t
    set the worst rank found so far r_t* to 0
    identify the Pareto Set P_t* of the population P_t
    while P_t* is not empty
        if r_t*==0
            rank individuals P_t* according to PO_k
            update r_t*
        else
            rank individuals P_t* according to PE
        end
        identify the Pareto Set P_t* of P_t = P_t – P_t*
```

```
        update r_i*
    end
end
```

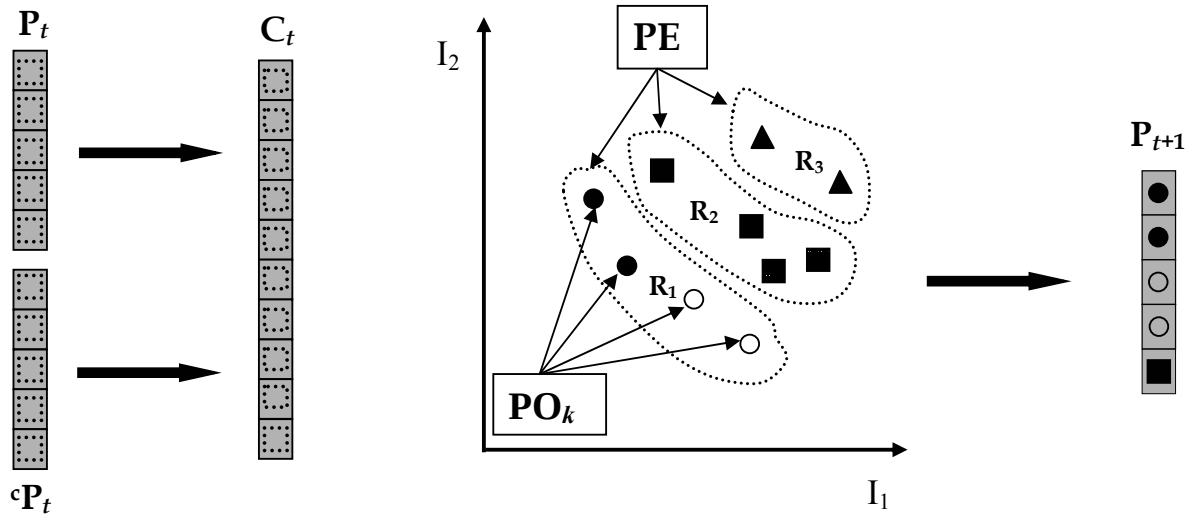

Figure 5: Ranking procedure of POGA. The Pareto efficient individuals (R_1) of C_t, are ranked through PO_k (the darker the shade, the better the individual). In this example, after all individuals in R_1 are inserted into P_{t+1}, the crowding distance is used to identify the individuals of rank 2 (R_2) that will complete the next generation population.

The advantage of this procedure over the standard NSGA-II is evident. When two solutions are assessed to select which will be either part of the mating pool (selection stage) or of the next generations' population (composition stage), their objective and constraint values are first translated into ranks and compared (we chose to implement for POGA and NSGA-II the scheme Constraint-First-Objective-Next, CFON). If this is not enough to state the winner, then the ranking scheme resorts to non rank-based information. NSGA-II and POGA adopt crowding distance as such a measure, which relates to the sparseness of solutions in the objective space. Needless to say, the more solutions present the same rank, the more often such non rank-based information determines the best solution. Since PO_k refines the rank of equally Pareto efficient solutions, the number of comparisons resolved by rank values is in favor of POGA.

Das (1998) claims and proofs that:

Claim 2 (C2). If $\boldsymbol{x}^*$ is efficient of order k, than it is efficient of order k+1.

In addition, we submit that:

Claim 3 (C3). If $\boldsymbol{x}^*$ is not efficient of order k, than it is not efficient of order k-1.

The proof immediately follows from C2. By induction, it is clear that if $\boldsymbol{x}^*$ is not efficient of order k, than it is not efficient of order j, for any $j < k$. The algorithm that we propose to compute the order of efficiency of a set of solutions to an MOP is based on *C3* and it is described by the following pseudo code:

```
set the rank of every solution j ∈ P_t rank_j = 0
set i = 0
set the flag isefficient_j = 0 for every solution j
for every (m-i)–element subset of the m objectives of the MOP
   for every solution j
      if isefficient_j == 1
         break
      if j is dominated by any other solution ∈ P_t for any of the ^mC_{m-i} (m-i)–element subsets
         do not update rank_j and change the flag of j isefficient_j = 1
      else
         rank_j = rank_j + 1
      end
   end
   i = i +1
end
```

If researchers have successfully applied PO as a post processing routine in the attempt to further rank the set of the best solutions found at the end of the search, to the authors' best knowledge POGA is the first algorithm that embodies PO in the ranking procedure.

TEST FUNCTION

To assess the behaviour of the Algorithm presented and to compare it to NSGA-II we chose the test function DTLZ2 [11], which is often used to tests the scalability of MOGAs [1]. DTLZ2 is constructed as follows:

$$\begin{cases} \min\limits_{\boldsymbol{x}\in\Omega} \boldsymbol{f}(\boldsymbol{x}) \\ \Omega = \{\boldsymbol{x} \mid 0 \le x_i \le 1 \; \forall i = 1,\ldots,n\} \\ f_1(\boldsymbol{x}) = (1+\boldsymbol{g}(\boldsymbol{x}_M)) \cdot \cos(x_1 \cdot \pi/2) \cdots \cos(x_{m-2} \cdot \pi/2) \cdot \cos(x_{m-1} \cdot \pi/2) \\ f_2(\boldsymbol{x}) = (1+\boldsymbol{g}(\boldsymbol{x}_M)) \cdot \cos(x_1 \cdot \pi/2) \cdots \cos(x_{m-2} \cdot \pi/2) \cdot \sin(x_{m-1} \cdot \pi/2) \\ f_3(\boldsymbol{x}) = (1+\boldsymbol{g}(\boldsymbol{x}_M)) \cdot \cos(x_1 \cdot \pi/2) \cdots \sin(x_{m-2} \cdot \pi/2) \\ \vdots \\ f_m(\boldsymbol{x}) = (1+\boldsymbol{g}(\boldsymbol{x}_M)) \cdot \sin(x_1 \cdot \pi/2) \\ \boldsymbol{g}(\boldsymbol{x}_M) = \sum_{x_i \in \boldsymbol{x}_M} (x_i - 0.5)^2 \end{cases}$$

where n is the number of decision variables, m is the number of objective functions and $\boldsymbol{x}_M$ are the last $M=(n-m+1)$ decision variables. The difficulty of this problem is represented by the huge number of local Pareto optimal fronts (11^M-1) that can attract the solutions generated by the MOEA before reaching the global Pareto optimal front.

To compare the performance of the two algorithms we adopted a set of metrics commonly used in the literature.

Two set Coverage (CS) [24]. This metric relates to the relative coverage of two sets. Consider X', $X'' \subseteq X'$ as two sets of points in the objective space of an MOP. CS is defined as

$$CS(X', X'') \triangleq \frac{|\{\boldsymbol{x''} \in X''; \exists\, \boldsymbol{x'} \in X' : \boldsymbol{x'} \preceq \boldsymbol{x''}\}|}{|X''|}$$

If all the points in *X'* dominate or are equal to all points in X'', then $CS(X', X'') = 1$; if all points dominate in X'' dominate the points in X', $CS(X', X'') = 0$. Since, in general $CS(X', X'') \neq 1 - CS(X'', X')$, both values should be considered.

Generational Distance (GD) [4]. This metric relates to the average distance of a set of points from a Reference Set (RS) and is defined as

$$GD \triangleq \frac{(\sum_{i=1}^{n} d_i^2)^{1/2}}{n}$$

where *n* is the number of points in the set, *d* is the Euclidean distance between each point in the set and its nearest RS-point.

Normalized Hyper Area (Hn). This metric relates to the sparseness of a set of points and is a variation of the one used by [22]. It measures the fraction of the space between the axes origin and a reference point that is dominated by the members of the set. It is defined as

$$Hn \triangleq \frac{\bigcup_{i=1}^{n^*} hv_i}{hv_O}$$

where n^* is the number of non-dominated points in the set, hv_i is the *hyper*-volume enclosed between the point *i* and the Reference Point (RP) and hv_O is the *hyper*-volume enclosed between the origin and the reference point.

Results

In order to assess the scaling behaviour of POGA and NSGA-II, the algorithms were applied to the optimization of DTLZ2 with an increasing number of objective functions. Since, according to Claim *C1*, the difference between Pareto Optimality and Preference Ordering ranking procedures is almost negligible for a three-dimensional MOP, it was decided to focus the analysis on four, five and six objectives, i.e. $m = 4,5$ and 6 respectively. The value of *M* was set to 10 for all the three tests; consequently, *n* varied from 13 to 15. Both POGA and NSGA-II used a binary encoding despite some authors suggesting a real variable representation, because a better parameter tuning was achieved. The population was set to 100 individuals for all the experiments. This is in contrast to the methodology used by Branke [1] where the population size was varied with the number of objective functions to maintain a somewhat constant proportion of non dominated-solutions within the population, following some empirical findings provided in Deb [9]. The aim was to limit as much as possible the noise when comparing the behaviour of different algorithms. The choice made in the present study is motivated by the fact that the implicit purpose of the test presented herein is to compare the efficiency of two optimality conditions. The single point cross over and uniform mutation were performed at the rate of .6 and .01 respectively.

Table 1 summarises the statistics of the metrics GD and Hn for the two algorithms considered. To compute GD, the RS was formed by generating 1000 points uniformly distributed over the true Pareto front, which will be referred to as $\mathcal{PF}^*_{true}$. To compute Hn, the points [1,1,1,1], [1,1,1,1,1] and [1,1,1,1,1,1] were chosen as the RP for $m = 4,5$ and 6 respectively.

Table 1: Performance values of NSGA-II and POGA applied to the optimization of three implementations of the test function DTLZ2, i.e. is with *m* set to 4, 5 and 6 respectively.

			NSGA-II	POGA
GD	***m* = 4**	*Mean*	0.2487	0.2576
		Std	0.0159	0.0187
	***m* = 5**	*Mean*	0.3288	0.3053
		Std	0.0282	0.0139
	***m* = 6**	*Mean*	0.6449	0.5178
		Std	0.0601	0.0315
Hn	***m* = 4**	*Mean*	0.5923	0.5941
		Std	0.0072	0.0073
	***m* = 5**	*Mean*	0.6403	0.6471
		Std	0.0072	0.0130
	***m* = 6**	*Mean*	0.6350	0.6820
		Std	0.0191	0.0091

As the values of GD suggest, after 200 generations all the MOGA converged fairly well to $\mathcal{PF}_{true}$. It is interesting to note that as the number of objectives (*m*) increases, POGA performs progressively better than NSGA-II, as both GD and Hn values clearly show.

Figure 6 shows a pictorial representation of the CS-metric values for two algorithms. Each graph represents a box plot of the CS values computed on the last generation solutions of each run for *m*=4,5 and 6 respectively. The plot at the top right relates to CS(POGA,NSGA-II), while the plot at the bottom left to CS(NSGA-II, POGA). As the number of objectives (*m*) increases, a progressively higher number of solutions found by NSGA-II are dominated by those found by POGA; conversely, a progressively smaller number of those found by POGA are dominated by those generated by NSGA-II.

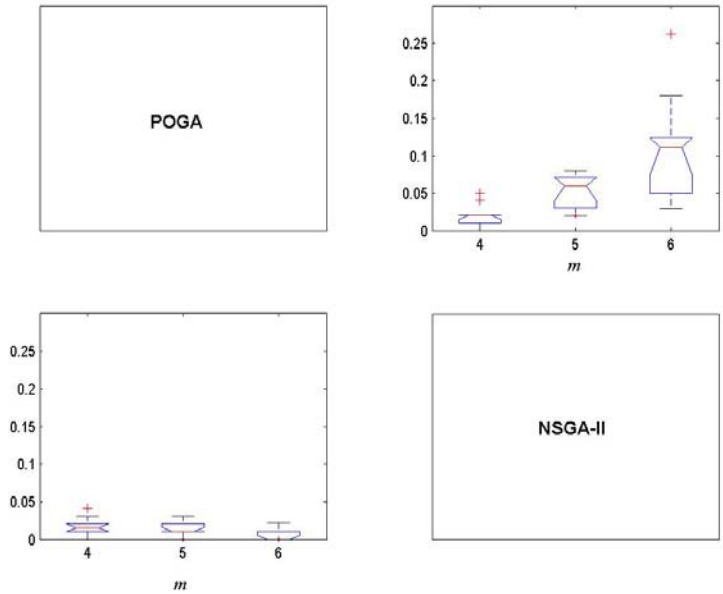

Figure 6: Box-plot of the CS values for the three MOGA applied to DTLZ2. The graph (*i*,*j*) represents the statistics of CS(*i*,*j*) on the 10 runs performed for *m* = 4,5 and 6.

This pattern supports the argument that the ranking procedure adopted by POGA provides it with a strong advantage over conventional Pareto-based MOGA when dealing with highly-dimensional MOPs.

Figure 7 depicts the generation-wise Pareto efficiency plots of a run for the three implementations of the function DTLZ2, i.e. for m = 4,5 and 6. At each generation, solutions found by the two algorithms are grouped and assessed through Pareto efficiency and individuals dominated are represented by dots. The individuals found by NSGA-II are grouped in the upper half of each plot, while those found by POGA in the lower half. It is interesting to note that, after a few generations where the two algorithms perform in a similar way, POGA takes over and consistently finds better solutions.

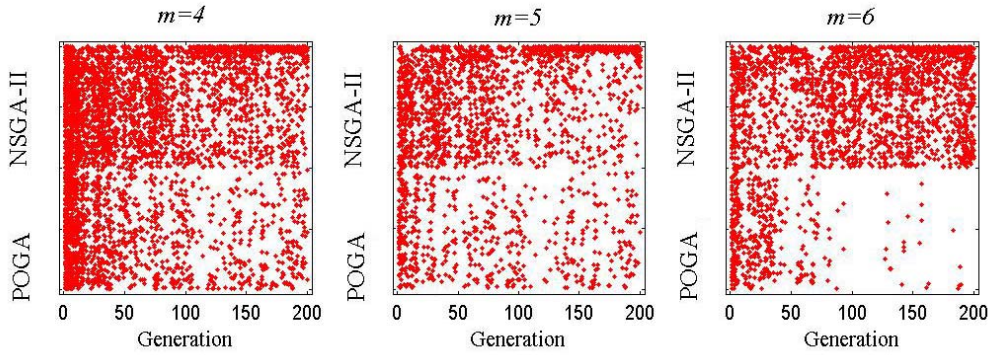

Figure 7: Generation-wise Pareto efficiency plot for each of the three implementations of the test function DTLZ2 (m = 4,5,6).

AUTOMATIC MODEL CALIBRATION

The Preference Ordering Genetic Algorithm was tested on the calibration of BEMUS, a storm sewer network, physically-based, distributed model. This section is organised as follows: first, the description of the model, the catchment, and the problem of the automatic model calibration are given. Then, the results of the application of POGA are presented for the calibration and validation data sets. Finally the performance of POGA is compared to that of NSGA-II.

Description of the model

The described algorithm was tested on the calibration of a storm sewer network model using experimental rainfall-runoff data. The BEMUS simulation model was used, which is a physically-based, distributed model. It has two components: (i) computation of surface runoff hydrographs, and (ii) simulation of flow in the sewer network. The methodology applied in the first ("hydrologic") component and the original version of the model were described by Radojković and Maksimović [19]. An updated version with improved pipe flow simulation was developed by Djordjević [12].

Catchment decomposition. The catchment is divided into smaller areas – subcatchments – that are drained into particular sewer network nodes. Each subcatchment is further divided into impervious (roofs, streets, etc.) and pervious parts (parks, gardens, etc.). Runoff from each of these surfaces is calculated separately (in accordance with their areas), summed up, and then introduced into the network nodes along with the dry weather flows.

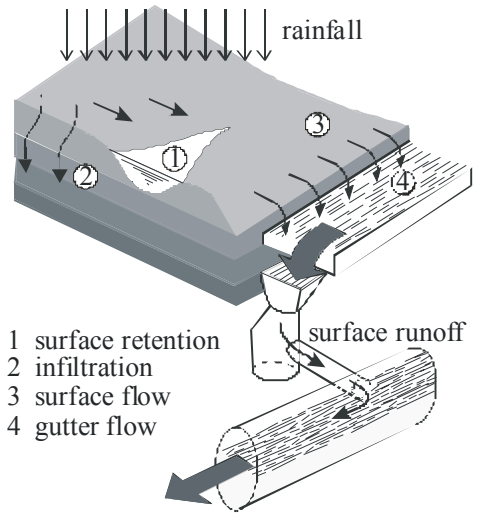

Figure 8: Surface runoff phases

Surface retention. Filling up of small surface holes (phase 1 in Fig. 8) is calculated via the Linsley empirical formula:

$$h_{\mathrm{e}}(t) = h_i(t) - h_{\mathrm{d}}\left[1 - \exp(-h_i/h_{\mathrm{d}})\right]$$

where h_{e} = effective rainfall depth at time t, h_i = total rainfall depth and h_{d} = surface-dependant empirical parameter denoting the equivalent thickness of the layer of water equally distributed over the entire pervious or impervious area.

Infiltration. By combining mass and momentum conservation equations for vertical infiltration in unsaturated porous media, and assuming that the soil moisture at the beginning of rainfall is known, an ordinary differential equation is obtained, which is solved analytically. In doing so, it is necessary to know the time between the beginning of the rainfall and the moment of saturation of surface soil, which is determined numerically from the momentum conservation equation written in the integral form. This approach is commonly referred to as Green-Ampt modification of the Richards equation, that reads:

$$H = H_{\mathrm{s}} + h_{\mathrm{c}} \ln\left(\frac{h_{\mathrm{c}} + H}{h_{\mathrm{c}} + H_{\mathrm{s}}}\right) + \frac{k_{\mathrm{s}}}{\varepsilon_0}(t - t_{\mathrm{s}})$$

where H = depth of the wet front reached at time t, H_{s} = wet front reached at time t_{s}, t_{s} = time from the beginning of rainfall to total surface soil saturation, h_{c} = absolute value of referential capillary height, k_{s} = Darcy coefficient for saturated soil and ε_0 = effective porosity. By subtracting the infiltration on pervious areas and surface retention on all areas from the rainfall intensity, the net (effective) rainfall i_{e} is obtained. Other hydrologic losses (e.g. evapotranspiration) are considered to be negligible, which can be justified for short-term events.

Surface flow and gutter flow. Each subcatchment area is replaced by two equivalent rectangular areas with constant slopes, from which the water flows to the gutter (as shown in Fig. 9). This way the 2D overland flow is replaced by two 1D flows. 1D schematized surface sheet flow is described by mass and momentum conservation equations. By neglecting some terms and after certain transformations, these are reduced to kinematic-wave equations:

$$\frac{\partial h}{\partial t}+\frac{\partial q}{\partial x}=i_{\mathrm{e}}$$

$$-ghI_0+\frac{\tau_{\mathrm{b}}+\tau_{\mathrm{i}}}{\rho}=0$$

where h = flow depth, q = discharge per unit width, x = coordinate in the flow direction (perpendicular to the gutter in this case), I_0 = bed slope, g = gravitational acceleration, ρ = water density, τ_{b} = bed shear stress and τ_{i} = additional shear stress due to the influence of the inertial force of rain drops. By supposing that the water level profile is of the parabolic shape, these equations are reduced to an ordinary differential equation, which is solved by the Euler modified method. Gutter flow is described with the analogous equations (by taking lateral inflow instead of effective rainfall), and solved in a similar way.

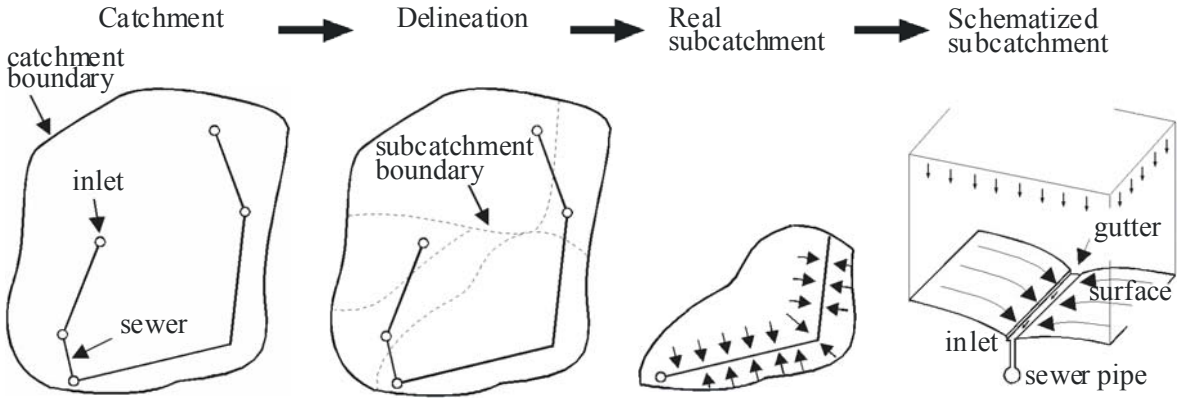

Figure 9: Catchment schematization

Source control. It is possible to define a source control technique, i.e. which portion of roofs (p_{roof}) and of other impervious areas (p_{imp}) is drained directly to the sewer system, and from which portion of these areas the water is spilled on the pervious areas (where it is partly infiltrated) prior to reaching the sewer system. Values of these two parameters, which can be between 0 and 1, crucially influence the runoff volume.

Sewer network flow. Muskingum-Cunge Variable Parameter method [18] is applied for solving kinematic-wave equations to simulate flow in sewer pipes:

$$Q_{i+1}^{j+1}=C_1Q_i^j+C_2Q_i^{j+1}+C_3Q_{i+1}^j$$

$$C_1=\frac{2\psi+Cr}{2(1-\psi)+Cr}\qquad C_2=\frac{-2\psi+Cr}{2(1-\psi)+Cr}\qquad C_3=\frac{2(1-\psi)-Cr}{2(1-\psi)+Cr}$$

where Q = discharge, ψ = spatial weighting coefficient and Cr = Courant number. At each step, ψ and Cr are iteratively calculated as averages of local values at four computational points in x-t plane (covering cross-sections x_i and $x_{i+1}=x_i+\Delta x$ and time levels t^j and $t^{j+1}=t^j+\Delta t$), as $Cr=c\Delta t/\Delta x$ and $\psi=(1-Q/\Delta xI_0Bc)/2$, where $c=dx/dt=dQ/dA$ = wave celerity, A = cross-sectional area and B = water table width. By such a choice, numerical parameters are linked to the physical parameters (pipe roughness, bed slope and current cross-section geometry) so that

numerical diffusion of the finite difference solution of kinematic wave equations matches the actual diffusion which would be obtained by the more complete diffusion wave equations. Pipe roughness is introduced through the assumption (inherent to kinematic wave model) that the friction slope is equal to the bottom slope.

The described pipe flow model is applicable to networks in which backwater effects are negligable and surcharging is rare (steep, not overloaded systems), otherwise the full dynamic model is required. The kinematic wave model is adopted here beacuse it is much faster and yet reasonably accurate for the catchment and the events described in the sequel.

Description of the catchment

The Experimental urban catchment Miljakovac (in Belgrade, Serbia) covers 25.5ha, 10.5% of which are roofs and further 27.3% are other impervious areas (Fig. 10a). Storm sewer network consists of 112 pipes, diameters ranging from 300 to 600mm, total length 4.1km. Two flow gauges recorded the runoff, one from the south-west part of the catchment and one from the entire area.

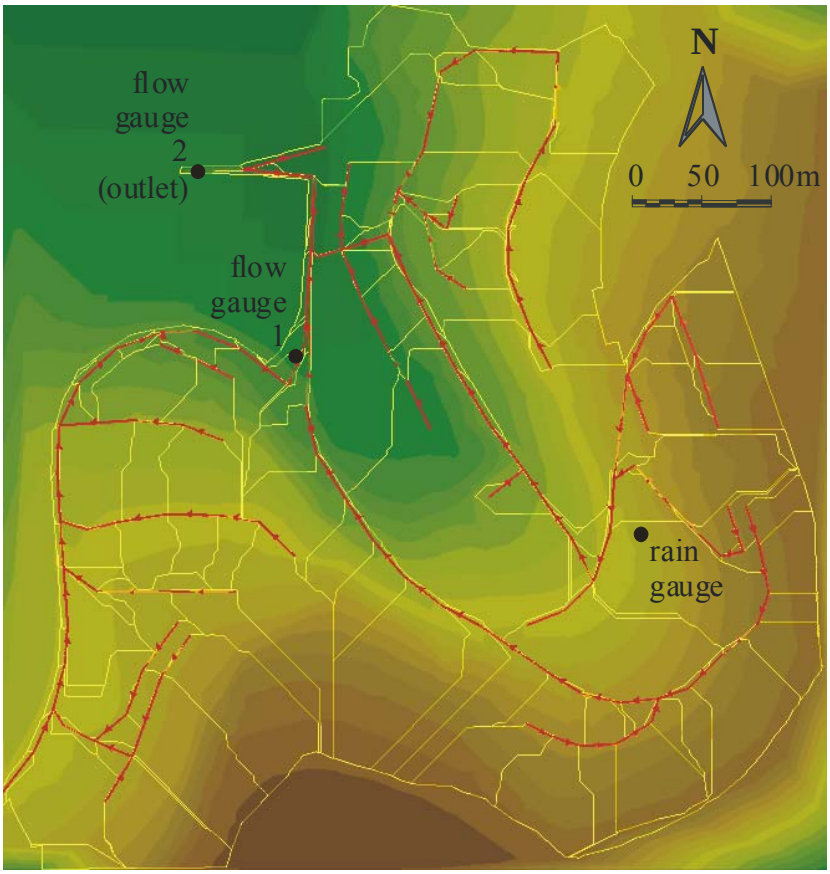

Figure 10: Miljakovac catchment

Problem formulation

The problem of the multi-objective calibration of a model can be generally stated as follows [17]

$$\min_{\boldsymbol{\theta}\in\Omega} \boldsymbol{f}(\boldsymbol{\theta})$$

where $\boldsymbol{f}$ is a m-dimensional vector-valued objective function, $\boldsymbol{\theta}$ is the model parameter set and Ω is the feasible parameter space. For this study, the following functions were chosen to measure the agreement between model-simulated and observed outflows through:

Overall volume error. $$f_1(\boldsymbol{\theta}) \triangleq \sum_{i=1}^{n}\left|\widehat{Q}_i - Q_i(\boldsymbol{\theta})\right| / (\sum_{i=1}^{n}\widehat{Q}_i)$$

Overall root mean square error (RMSE). $$f_2(\boldsymbol{\theta}) \triangleq \left[\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}_i - Q_i(\boldsymbol{\theta})\right]^2\right]^{1/2}$$

Peak flow error. $$f_3(\boldsymbol{\theta}) \triangleq \left|\max \widehat{Q} - \max Q(\boldsymbol{\theta})\right|$$

Time to peak. $$f_4(\boldsymbol{\theta}) \triangleq t_{\arg\max \widehat{Q}} - t_{\arg\max Q(\boldsymbol{\theta})}$$

where $\widehat{Q}$ and $Q(\boldsymbol{\theta})$ represent the measured and simulated flow respectively and n is the number of observations. Since measurements from two gauging stations were available, it was decided to take into accounts 8 objective functions altogether for the purpose of calibration. In the remainder of the paper, ${}^{u}f_i(\boldsymbol{\theta})$ and ${}^{d}f_i(\boldsymbol{\theta})$ denote the objective function i for i=1,2,34 computed at the up-stream or downstream gauging stations respectively. Table 2 summarizes the parameters of the model BEMUS that are to be calibrated.

Table 2: Parameter vector θ for the model BEMUS.

PARAMETER NAME	DESCRIPTION	LOWER LIMIT	UPPER LIMIT
P1	Retention capacity of pervious area (mm)	.8	4
P2	Retention capacity of impervious area (mm)	.4	1
P3	Darcy coefficient (m/s)	.000001	.00000001
P4	Soil porosity	0.3	0.55
P5	Sub-catchment shape factor	0.1	10
P6	Pervious area Manning factor	0.01	0.50
P7	Impervious area Manning factor	0.01	0.05
P8	Roofs drained to sewer system	0	1
P9	Impervious areas drained to sewer system	0	1
P10	Manning factor for pipes (*)	0.01	0.02

(*) All pipes in the sewer network are assigned the same Manning factor.

Calibration

The calibration of the model BEMUS was undertaken with POGA. Table 3 shows the main features of the rainfall event that was chosen for this purpose.

Table 3: Details of the rainfall event used for the calibration of the model BEMUS. The number of observations, the sampling frequency and the type of event (Single-Burst/Multi-Burst) is reported.

Event Name	No. Observations (#)	Sampling Freq. (#/h)	Event Type
Yu01	111	60	SB

For this application, the eight parameters of the simulation model were encoded as binary variables. After some trials with different selection and recombination operators, it was decided to implement the bit-wise tournament selection, single point cross over and uniform mutation. The population size, probability of crossover and mutation were set to 200, 0.6 and 0.005 respectively; the maximum number of generations per run was set to 100. Given the stochastic nature inherent in evolutionary algorithms, ten different runs were performed with POGA in order to take into account the effect of different random number generator seeds. The results are presented for the set S^*_{POGA}. This set is obtained by first taking the individuals efficient to the lowest order of each run performed with POGA and grouping them into S_{POGA}; then Pareto efficiency is performed on this set and only the non-dominated individuals are retained and grouped to form S^*_{POGA}.

Table 4 summarizes the performances of the set S^*_{POGA}. In order to infer the maximum information from the solutions found, S^*_{POGA} was ranked through Preference Ordering, by looking at both orders and degrees of efficiency of each set of parameters ($PO_{k,z}$). The results are then presented for group of solutions of the same rank and for group of ranks. S^*_{POGA} consists of 665 Pareto efficient solutions. There are 373 solutions efficient of order 8; 291 are efficient of order 7 and there is a single best solution (rank I), that is efficient of order 6. It should be noted that this solution is the best compromise amongst the entire set S^*_{POGA} and therefore, it represents an interesting parameter set from an engineering point of view. We denote this solution by ${}^{+}S^*_{POGA}$.

Table 4: Mean and standard deviation of the objective values of the solutions $\in$ S*POGA. The statistics are presented for solutions grouped by rank (roman letters) and group of ranks; the number of solutions in each group is also given. Orders of efficiency are shown in the adjacent brackets. The last two groups are formed by collecting together individuals from rank I and II and from Rank I to III respectively.

Rank	No. of points	Stat	${}^{u}f_1(\theta)$	${}^{d}f_1(\theta)$	${}^{u}f_2(\theta)$	${}^{d}f_2(\theta)$	${}^{u}f_3(\theta)$	${}^{d}f_3(\theta)$	${}^{u}f_4(\theta)$	${}^{d}f_4(\theta)$
I (6)	1	*Mean*	0.127	0.152	0.003	0.015	0.005	0.000	0.000	0.000
		Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
II (7)	291	*Mean*	0.166	0.104	0.004	0.011	0.003	0.023	0.000	0.000
		Std	0.025	0.025	0.001	0.002	0.002	0.009	0.000	0.000
III (8)	373	*Mean*	0.153	0.122	0.004	0.013	0.002	0.022	0.000	0.000
		Std	0.021	0.020	0.000	0.002	0.002	0.010	0.000	0.000
{I-II}	292	*Mean*	0.165	0.104	0.004	0.011	0.003	0.022	0.000	0.000
		Std	0.025	0.025	0.001	0.002	0.002	0.009	0.000	0.000
{I-III}	665	*Mean*	0.158	0.114	0.004	0.012	0.002	0.022	0.000	0.000
		Std	0.024	0.024	0.001	0.002	0.002	0.010	0.000	0.000

For the remainder of the analysis, we restricted our attention to the set ${}^{E}S^*_{POGA}$, i.e. the subset of S^*_{POGA} that comprises solutions of rank I and II. Table 5 summarizes the parameter values of ${}^{E}S^*_{POGA}$ and ${}^{+}S^*_{POGA}$.

Table 5: Mean and standard deviation of parameter values for the set ${}^{E}S^{*}_{POGA}$, that is the set formed by grouping together the individuals of S^{*}_{POGA} of rank 1 and 2; the parameter values of ${}^{+}S^{*}_{POGA}$, the best solution in ${}^{E}S^{*}_{POGA}$ (i.e. the only efficient of order 6) are also presented.

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
${}^{E}S^{*}_{POGA}$	*Mean*	2.4887	0.40447	0.00705	0.41408	0.90181	0.23262	0.04259	0.67551	0.69146	0.01854
	Std	0.87129	0.00748	0.00228	0.07212	0.78126	0.12204	0.01014	0.19512	0.11808	0.00218
${}^{+}S^{*}_{POGA}$		2.5668	0.4	0.00875	0.33235	0.33205	0.43575	0.05	0.98436	0.47703	0.02

As previously introduced, ${}^{+}S^{*}_{POGA}$ is an interesting solution from a practical point of view. It is, in effect, the solution amongst those collectively found by POGA that strikes the best balance of objective function values. Therefore it was decided to present the hydrographs obtained by routing the calibration rainfall data with the BEMUS model for this particular parameter set (Figure 10).

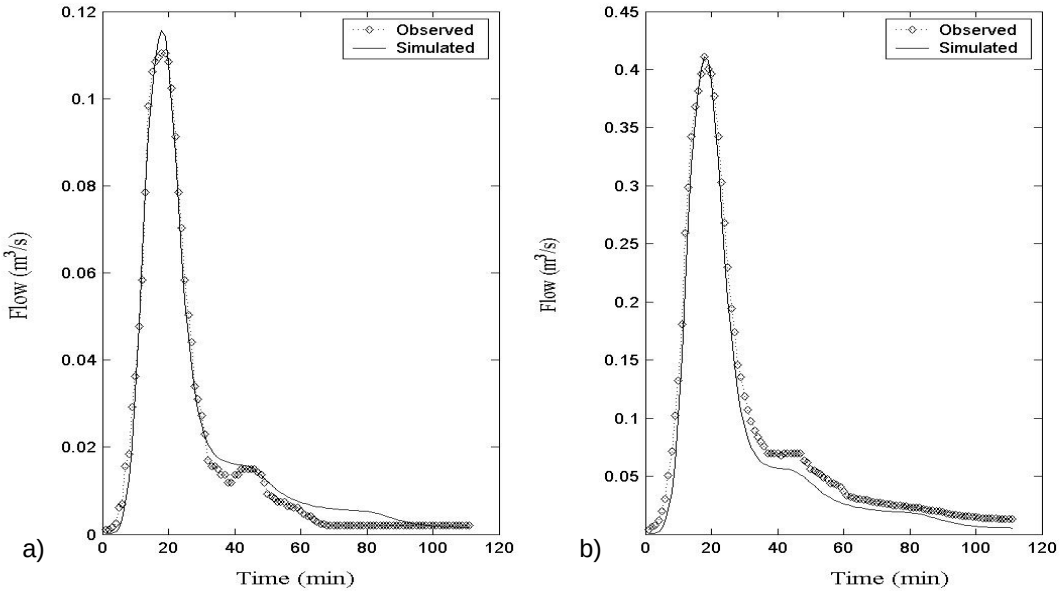

Figure 10: Observed and simulated flow generated ${}^{+}S^{*}_{POGA}$ on the calibration data set. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

Validation

To assess the quality of the calibration performed by POGA, the parameter sets in ${}^{E}S^{*}_{POGA}$ were simulated in the face of seven rainfall events, whose main features are summarized in Table 6.

Table 6: Details of the rainfall events used for the validation of the BEMUS model. The number of observations, the sampling frequency and the type of event (Single-Burst/Multi-Burst) is reported.

	Yu02	***Yu03***	***Yu04***	***Yu05***	***Yu06***	***Yu07***	***Yu08***
No. Observations (#)	51	46	121	221	101	171	101
Sampling Freq. (#/h)	12	30	30	30	30	30	60
Event Type	MB	SB	MB	MB	SB	MB	SB

The events chosen differ quite substantially and therefore should constitute an objective validation test suite. The results of the validation process are presented in Table 7, which summarizes the

performances of the parameters sets in ${}^{E}S^{*}_{POGA}$ measured in terms of the eight objective functions previously introduced.

Table 7: Validation results: statistics on the objective function values of the parameter sets in ${}^{E}S^{*}_{POGA}$ are given for the four events considered.

		${}^{u}f_1(\theta)$	${}^{d}f_1(\theta)$	${}^{u}f_2(\theta)$	${}^{d}f_2(\theta)$	${}^{u}f_3(\theta)$	${}^{d}f_3(\theta)$	${}^{u}f_4(\theta)$	${}^{d}f_4(\theta)$
Yu02	Mean	0.113	0.139	0.003	0.015	0.001	0.010	120	120
	Std	0.011	0.013	0.000	0.002	0.001	0.005	0	0
Yu03	Mean	0.145	0.283	0.002	0.010	0.002	0.011	3.637	1.171
	Std	0.019	0.038	0.000	0.001	0.002	0.004	0.790	0.987
Yu04	Mean	0.235	0.730	0.001	0.011	0.003	0.007	0	0
	Std	0.015	0.064	0.000	0.001	0.001	0.003	0	0
Yu05	Mean	0.356	0.458	0.005	0.020	0.005	0.038	0	0
	Std	0.030	0.068	0.000	0.003	0.001	0.006	0	0
Yu06	Mean	0.476	0.503	0.004	0.015	0.004	0.023	0.664	0.808
	Std	0.031	0.062	0.000	0.002	0.001	0.006	0.944	0.983
Yu07	Mean	0.618	0.609	0.005	0.020	0.008	0.035	0.308	2
	Std	0.037	0.072	0.000	0.002	0.002	0.006	0.723	0
Yu08	Mean	0.569	0.311	0.008	0.026	0.025	0.063	0.603	1.582
	Std	0.072	0.043	0.001	0.003	0.002	0.006	0.568	0.494

Validation results for the event *Yu02* show a poor value of the function that measures the time lag between recorded and simulated peak flow. Figure 11 shows observed and simulated hydrographs generated by the parameter set ${}^{+}S^{*}_{POGA}$ for this particular validation event. The observed hydrographs clearly show two distinct peaks that occurred about 150 minutes apart; although very similar, the second records a slightly higher flow.

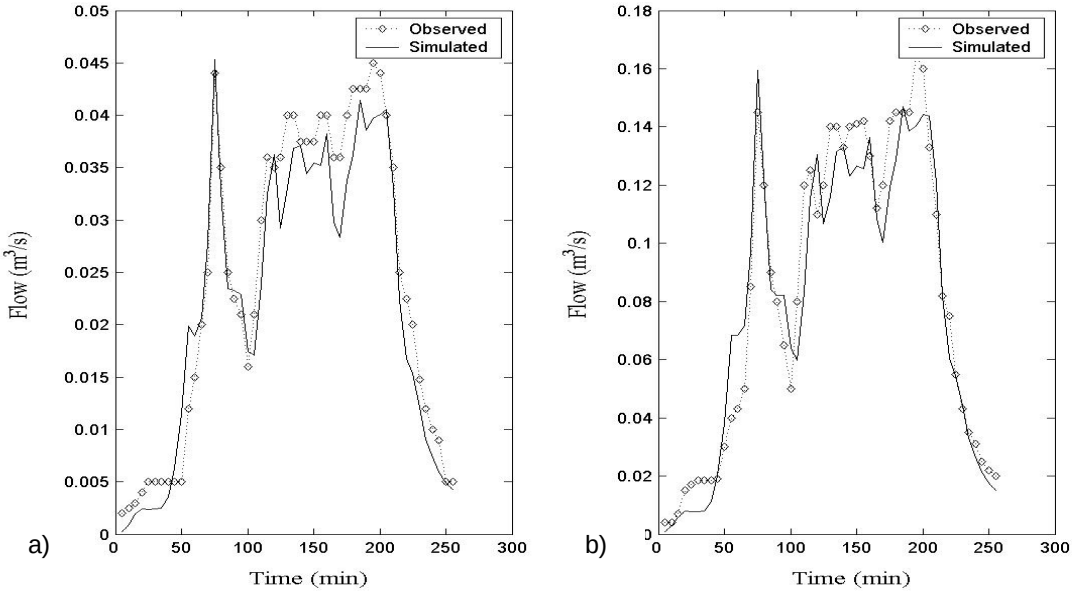

Figure 11: Observed and simulated flow generated ${}^{+}S^{*}_{POGA}$ on the validation data set *Yu02*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

The simulated hydrographs also show these two peaks but their relative importance is reversed compared to the measured ones. As a consequence, for this particular rainfall event, the value of the objective functions ${}^{u}f_4(\theta)$ and ${}^{d}f_4(\theta)$ do not measure the time elapsed between an observed peak flow and the same peak flow simulated. Instead, they measure the time elapsed between two different peak flows.

Figure 12 and 13 show the simulated and observed hydrographs for the validation events *Yu03* and *Yu05*.

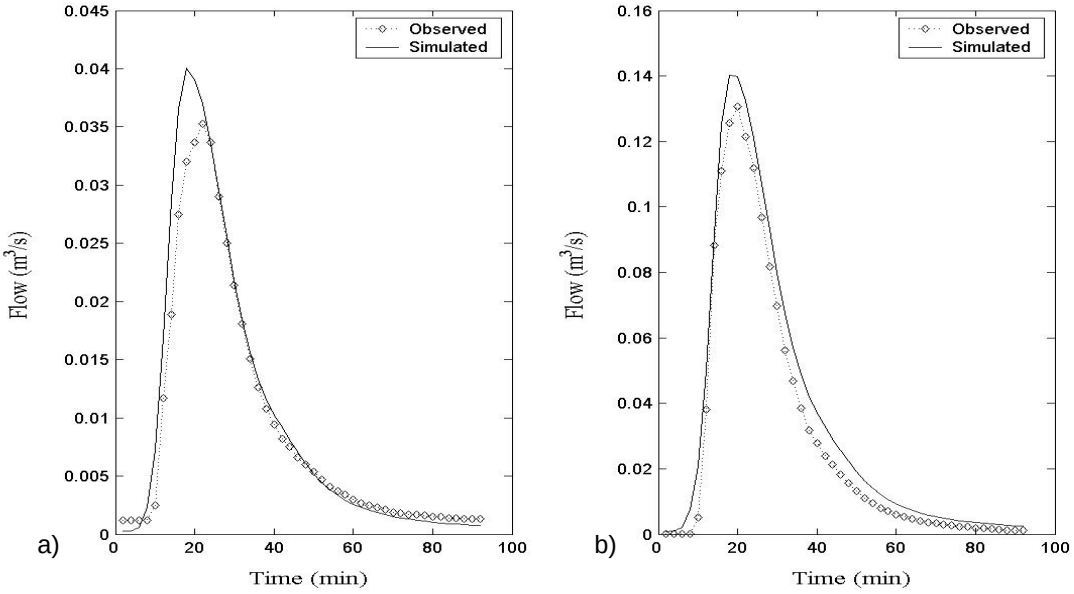

Figure 12: Observed and simulated flow generated ${}^{+}S^{*}_{\mathrm{POGA}}$ on the validation data set *Yu03*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

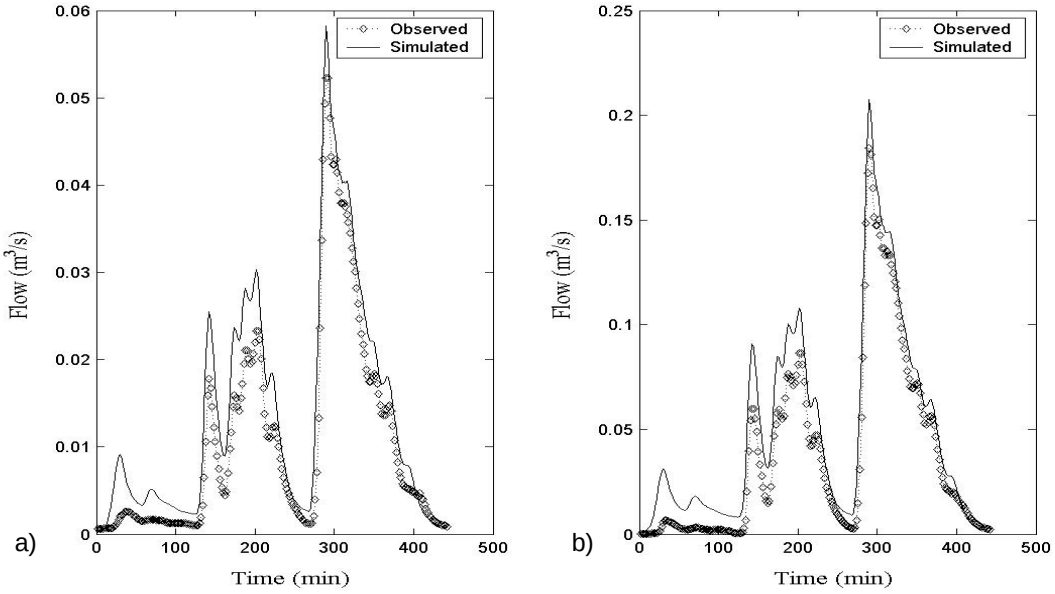

Figure 13: Observed and simulated flow generated ${}^{+}S^{*}_{\mathrm{POGA}}$ on the validation data set *Yu05*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

Discussion

To assess the performance of POGA, the calibration of the BEMUS model was also performed with NSGA-II. The encoding, the parameter setting and the number of runs were kept unchanged.

A preliminary analysis of the results showed the same pattern that emerged from the assessment of the CS metric presented for the test function DTLZ2. Figure 14 depicts the box plot of the CS metric for the last generation population of each of the ten runs performed with the two algorithms. As it can be appreciated, 5 to 65% of the parameter sets generated by NSGA-II are dominated by those generated by POGA. Conversely, only 0 to 0.5% of the solutions generated by POGA are dominated by those generated by NSGA-II.

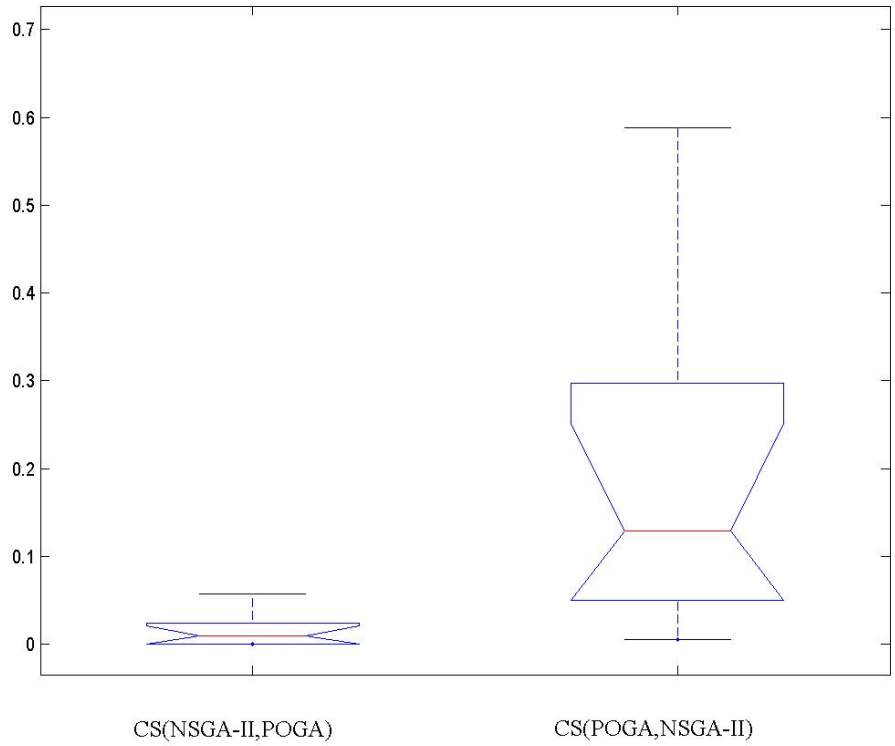

Figure 14: Box plot of the CS metric for the last generation population of each of the ten runs performed with POGA and NSGA-II.

The results of the calibration performed by NSGA-II are presented for the set $S^*_{NSGA\text{-}II}$. This set is obtained by first taking the individuals Pareto efficient of each run performed with NSGA-II and grouping them into $S_{NSGA\text{-}II}$. Then Pareto efficiency is performed on this set and only the non-dominated individuals are retained and grouped to form $S^*_{NSGA\text{-}II}$. Finally, $S^*_{NSGA\text{-}II}$ is sieved through by $PO_{k,z}$.: this not only permits to gain insights into the quality of these solutions, but it also allows to assess the effects of using Preference Ordering as a optimality condition embedded into a ranking scheme (POGA) rather than as a post processing routine (NSGA-II + PO). The results are then presented in Table 8 for groups of solutions of the same rank and for group of ranks. In details, when $S^*_{NSGA\text{-}II}$ was ranked through PO, seven ranks could be identified. None of the individuals falling into the six best-rank groups were Pareto efficient of order 6, i.e. not dominated by any other solution $\in S^*_{NSGA\text{-}II}$ for all the 28 combinations of 8 objectives taken 6 at the time. Since 84 individuals were efficient of order 7, they were further ranked according to their degree of efficiency. These resulted in the identification of 6 groups with different degrees of efficiency, ranging from 27 for individuals of rank I to 22 for those of rank VI. Therefore, individuals of rank I are those that lay on the boundaries of the set of individuals that are efficient of order 7.

Table 8: Mean and standard deviation of the objective values of the solutions $\in S^*_{NSGA\text{-}II}$. The statistics are presented for solutions grouped by rank (roman letters); the number of solutions comprised in each group is also given. Order and degree of efficiency are shown in the adjacent brackets. For instance, individuals of rank I are efficient of order 6 to the degree 27. The last two groups are formed by collecting together individuals from rank I to VI and from Rank I to VII respectively.

Rank	No. of points	Stat	${}^{u}f_1(\theta)$	${}^{d}f_1(\theta)$	${}^{u}f_2(\theta)$	${}^{d}f_2(\theta)$	${}^{u}f_3(\theta)$	${}^{d}f_3(\theta)$	${}^{u}f_4(\theta)$	${}^{d}f_4(\theta)$
I (6,27)	3	*Mean*	0.155	0.118	0.004	0.012	0.005	0.025	0.000	0.000
		Std	0.046	0.037	0.001	0.003	0.003	0.008	0.000	0.000
II (6,26)	19	*Mean*	0.158	0.114	0.004	0.012	0.004	0.025	0.000	0.000
		Std	0.035	0.029	0.001	0.002	0.002	0.009	0.000	0.000
III (6,25)	31	*Mean*	0.164	0.111	0.004	0.012	0.003	0.025	0.000	0.000
		Std	0.033	0.029	0.001	0.002	0.003	0.010	0.000	0.000
IV (6,24)	21	*Mean*	0.164	0.107	0.004	0.012	0.003	0.027	0.000	0.000
		Std	0.023	0.027	0.001	0.002	0.002	0.013	0.000	0.000
V (6,23)	4	*Mean*	0.180	0.088	0.004	0.011	0.002	0.032	0.000	0.000
		Std	0.011	0.008	0.000	0.001	0.001	0.008	0.000	0.000
VI (6,22)	6	*Mean*	0.158	0.112	0.004	0.013	0.002	0.028	0.000	0.000
		Std	0.004	0.011	0.000	0.001	0.001	0.012	0.000	0.000
VII (8)	685	*Mean*	0.184	0.124	0.005	0.015	0.003	0.024	0.050	0.034
		Std	0.042	0.039	0.001	0.004	0.003	0.022	0.217	0.180
{I-VI}	84	*Mean*	0.163	0.110	0.004	0.012	0.003	0.026	0.000	0.000
		Std	0.030	0.027	0.001	0.002	0.002	0.011	0.000	0.000
{I-VII}	769	*Mean*	0.182	0.122	0.005	0.014	0.003	0.024	0.044	0.030
		Std	0.042	0.038	0.001	0.004	0.003	0.021	0.206	0.170

It is now interesting to compare the quality of solutions in S^*_{POGA} with those in $S^*_{NSGA\text{-}II}$, rank by rank and order of efficiency by order of efficiency. The results of this analysis are presented in table 9. It is interesting to see that, none of the 665 solutions of the best three ranks ({I-III}) generated by POGA are dominated by any of the 53 solutions of the best three ranks generated by NSGA-II. Conversely, up to 20% of those generated by NSGA-II are dominated by those found by POGA. If we compare the two sets S^*_{POGA} and $S^*_{NSGA\text{-}II}$ by grouping solutions according to their order of efficiency, the gap between the performances of the two algorithms widens further. In fact, despite a maximum 0.15% of solutions found by POGA being dominated by those found by NSGA-II, up to 54% of solutions generated by NSGA-II are dominated. This result suggests that NSGA-II could only find a small number of relatively good solutions as opposed to POGA; in fact, the comparison between groups of the same order of efficiency entails many more NSGA-II solutions than that performed between groups of the same rank.

Table 9: CS values for the subsets obtained from $S^*_{NSGA\text{-}II}$ and S^*_{POGA} by grouping together individuals with the same rank or order of efficiency.

	Rank1	Rank2	Rank3	Order 6	Order 7	Order 8	Order{6,7,8}
CS(NSGA-II , POGA)	0	0	0	-	0	0.0027	0.0015
CS(POGA , NSGA-II)	0	0.1875	0.0645	-	0.3086	0.4571	0.5402

The analysis presented so far suggests that the set of best solutions found by POGA is overall qualitatively superior to those found by NSGA-II. Even though this represents a quite remarkable result, from an engineering point of view it is somewhat more interesting to focus on the comparison between the smallest subset of best solutions identified by the two algorithms; in fact, only a very limited number of parameter sets is amenable to further assessment by a modeler. According to Table 4, this subset for POGA consists only of the solution ${}^{+}S^{*}_{\text{POGA}}$. Similarly, according to Table 8, the subset for NSGA-II consists of the three solutions that have the best rank; we denote these solutions ${}^{1}S^{*}_{\text{NSGA-II}}$, ${}^{2}S^{*}_{\text{NSGA-II}}$ and ${}^{3}S^{*}_{\text{NSGA-II}}$. Table 10 shows their parameter values. As it can be seen, ${}^{+}S^{*}_{\text{POGA}}$ differs from the other solutions mainly for parameters P7, P8 and P9. It is also interesting to observe that the ${}^{1}S^{*}_{\text{NSGA-II}}$ value of P5 is significantly higher than that of the other best parameter sets.

Table 10: Parameter values of the highest rank-solutions found by POGA and NSGA-II.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
${}^{+}S^{*}_{\text{POGA}}$	2.5668	0.4	0.00875	0.33235	0.33205	0.43575	0.05	0.98436	0.47703	0.02
${}^{1}S^{*}_{\text{NSGA-II}}$	3.8523	0.4	0.00884	0.35392	2.3775	0.21904	0.04555	0.49658	0.83284	0.02
${}^{2}S^{*}_{\text{NSGA-II}}$	3.328	0.40059	0.00296	0.44706	0.36649	0.47411	0.04428	0.68524	0.65298	0.02
${}^{3}S^{*}_{\text{NSGA-II}}$	3.7148	0.40059	0.00873	0.36373	0.33023	0.27849	0.04238	0.85533	0.56305	0.02

A visual comparison of the hydrographs generated by the parameter sets ${}^{1}S^{*}_{\text{NSGA-II}}$, ${}^{2}S^{*}_{\text{NSGA-II}}$, ${}^{3}S^{*}_{\text{NSGA-II}}$ and ${}^{+}S^{*}_{\text{POGA}}$ is presented in Figures 15, 16, 17, and 18 for the calibration event *Yu01* and the three validation events *Yu02*, *Yu03* and *Yu05* respectively. It was purposely decided not to include the hydrographs generated by the parameter set ${}^{3}S^{*}_{\text{NSGA-II}}$ because they were undistinguishable from those generated by ${}^{2}S^{*}_{\text{NSGA-II}}$ for every rainfall event considered. Objective function values were consequently almost identical.

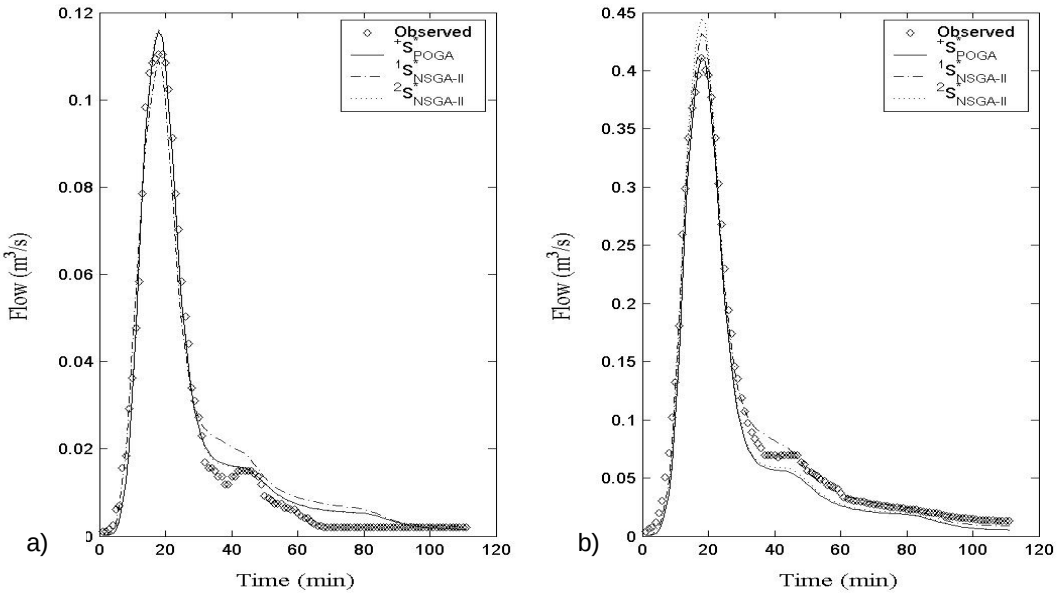

Figure 15: Observed and simulated flow generated by ${}^{1}S^{*}_{\text{NSGA-II}}$, ${}^{2}S^{*}_{\text{NSGA-II}}$ and ${}^{+}S^{*}_{\text{POGA}}$ on the calibration data set *Yu01*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

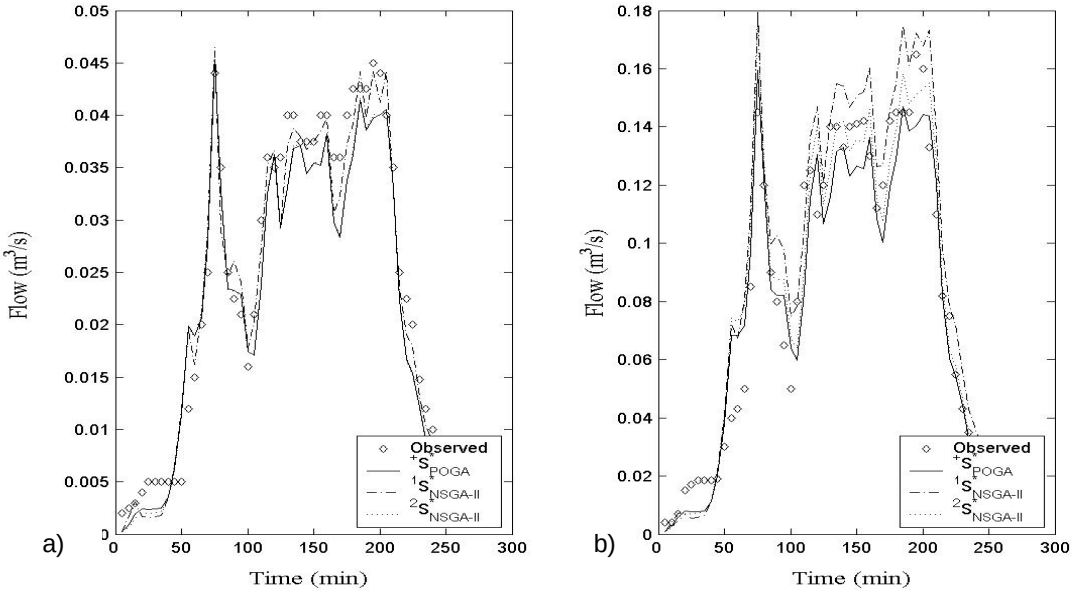

Figure 16: Observed and simulated flow generated by ${}^{1}S^{*}_{NSGA\text{-}II}$, ${}^{2}S^{*}_{NSGA\text{-}II}$ and ${}^{+}S^{*}_{POGA}$ on the validation data set *Yu02*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

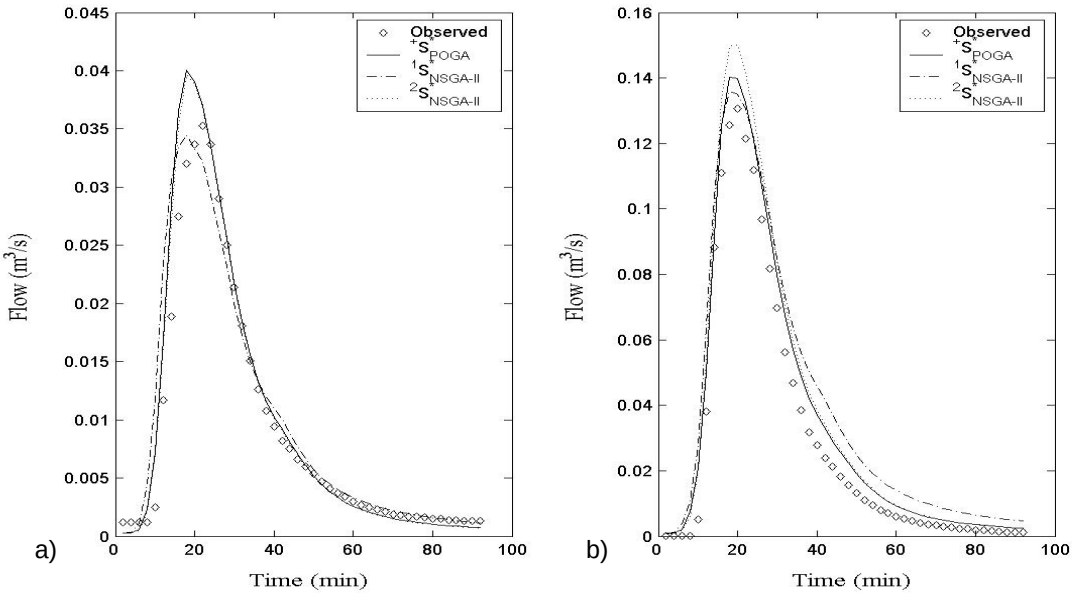

Figure 17: Observed and simulated flow generated by ${}^{1}S^{*}_{NSGA\text{-}II}$, ${}^{2}S^{*}_{NSGA\text{-}II}$ and ${}^{+}S^{*}_{POGA}$ on the validation data set *Yu03*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

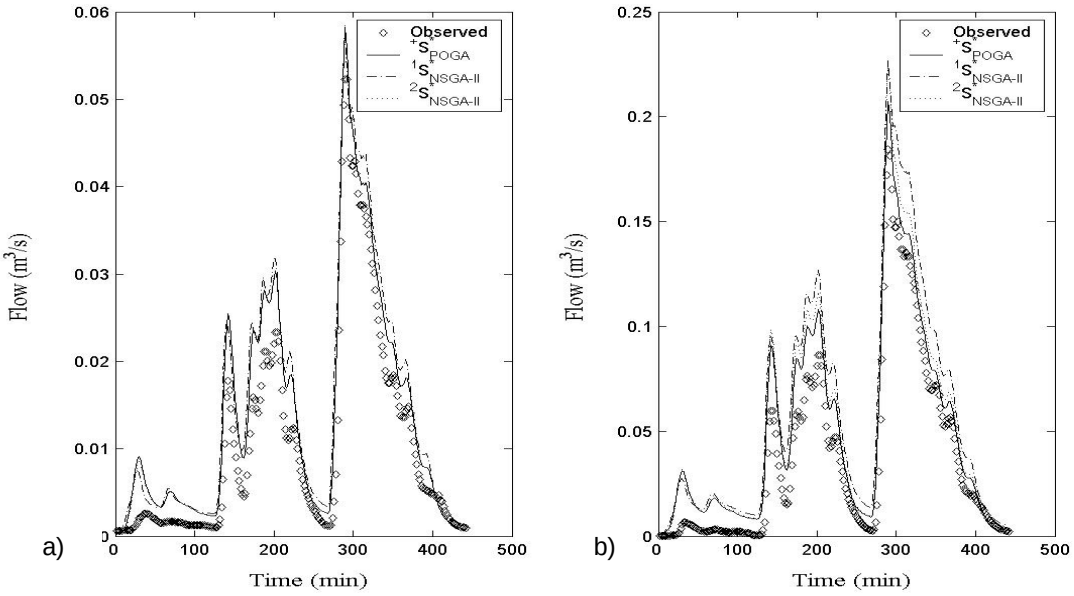

Figure 18: Observed and simulated flow generated by ${}^{1}S^{*}_{NSGA\text{-}II}$, ${}^{2}S^{*}_{NSGA\text{-}II}$ and ${}^{+}S^{*}_{POGA}$ on the validation data set *Yu05*. a) Hydrographs at the upstream gauging station; b) hydrographs at the downstream gauging station.

Albeit debatable, one might argue that from visual comparison, little can be stated about the quality of the hydrographs generated by POGA and NSGA-II-best solutions for the rainfall events *Yu01 Yu03* and *Yu05*. On the contrary, observing Figure 18 one can safely reason that ${}^{+}S^{*}_{POGA}$ produces hydrographs, both at the upstream and downstream gauging stations, that strike a better agreement with the observations than the hydrographs generated by NSGA-II best solutions.

To gain further insights into the quality of the 3 solutions ${}^{1}S^{*}_{NSGA\text{-}II}$, ${}^{2}S^{*}_{NSGA\text{-}II}$ and ${}^{+}S^{*}_{POGA}$, it was decided to compare their performances over the one calibration and seven validation events introduced previously. To this end, the objective function values generated by these solutions were grouped, event by event, and ranked according through $PO_{k,z}$. Table 11 summarizes the results of this analysis. As it can be appreciated, ${}^{+}S^{*}_{POGA}$ is the best solution for 4 out of a total of 8 events (*Yu02*, *Yu04*, *Yu05* and *Yu06*). For another 2 events ${}^{+}S^{*}_{POGA}$ shares the best rank with ${}^{1}S^{*}_{NSGA\text{-}II}$ (*Yu01*) and ${}^{2}S^{*}_{NSGA\text{-}II}$ (*Yu07*); on the remainder 2 calibration events (*Yu03* and *Yu08*), ${}^{+}S^{*}_{POGA}$ is the second-best parameter set. It is worth emphasizing that one of the best solutions generated by NSGA-II, namely ${}^{2}S^{*}_{NSGA\text{-}II}$, is not Pareto efficient when simulated in the face of validation events *Yu07* and *Yu08*.

These results support us interpreting the glimpses that emerged from the visual comparison. On the calibration event, ${}^{+}S^{*}_{POGA}$ performs equally to ${}^{1}S^{*}_{NSGA\text{-}II}$ in terms of the 8 objective functions considered. The other solution generated by NSGA-II shows a somewhat poorer calibration. On *Yu02* and *Yu03* (i.e. the first and second validation events considered), ${}^{+}S^{*}_{POGA}$ is the best and the second-best solution respectively. On the last validation event considered for visual comparison, *Yu05*, the situation appears somewhat different: not only ${}^{+}S^{*}_{POGA}$ is the best solution, but is efficient of order 4 while the others are only Pareto efficient. This implies that ${}^{+}S^{*}_{POGA}$ is the best solution for every combination of the 8 objective functions, taken 4 at the time. Among these combinations, there are $\{ {}^{u}f_1(\boldsymbol{\theta}), {}^{u}f_2(\boldsymbol{\theta}), {}^{u}f_3(\boldsymbol{\theta}), {}^{u}f_4(\boldsymbol{\theta})\}$ and $\{ {}^{d}f_1(\boldsymbol{\theta}), {}^{d}f_2(\boldsymbol{\theta}), {}^{d}f_3(\boldsymbol{\theta}), {}^{d}f_4(\boldsymbol{\theta})\}$: the former entails only the objective functions that relate to the upstream gauging station, while the latter only those that relate to the downstream one. As a consequence, from a visual comparison, both hydrographs generated by ${}^{+}S^{*}_{POGA}$ appear significantly better than those generated by ${}^{1}S^{*}_{NSGA\text{-}II}$ (*Yu01*) and ${}^{2}S^{*}_{NSGA\text{-}II}$.

Table 11: Objective function values and ranks of the solutions ${}^{1}S^{*}_{\text{NSGA-II}}$, ${}^{2}S^{*}_{\text{NSGA-II}}$ and ${}^{+}S^{*}_{\text{POGA}}$. Orders and degrees of efficiency are shown in the adjacent brackets.

Event		${}^{u}f_1(\theta)$	${}^{d}f_1(\theta)$	${}^{u}f_2(\theta)$	${}^{d}f_2(\theta)$	${}^{u}f_3(\theta)$	${}^{d}f_3(\theta)$	${}^{u}f_4(\theta)$	${}^{d}f_4(\theta)$	Rank
	${}^{+}S^{*}_{\text{POGA}}$	0.1271	0.1524	0.0030	0.0153	0.0053	0.0000	0	0	I (5,55)
Yu01	${}^{1}S^{*}_{\text{NSGA-II}}$	0.2081	0.0751	0.0051	0.0087	0.0014	0.0212	0	0	I (5,55)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.1279	0.1391	0.0031	0.0142	0.0059	0.0344	0	0	II (7)
	${}^{+}S^{*}_{\text{POGA}}$	0.1289	0.1329	0.0038	0.0139	0.0004	0.0055	24	24	I (5)
Yu02	${}^{1}S^{*}_{\text{NSGA-II}}$	0.0973	0.1690	0.0029	0.0177	0.0015	0.0145	24	24	II (7)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.1293	0.1335	0.0037	0.0143	0.0004	0.0105	24	24	III (8)
	${}^{+}S^{*}_{\text{POGA}}$	0.1517	0.1988	0.0027	0.0073	0.0047	0.0096	2	1	II (6)
Yu03	${}^{1}S^{*}_{\text{NSGA-II}}$	0.1761	0.3376	0.0032	0.0116	0.0008	0.0052	2	1	III (7)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.1394	0.2611	0.0024	0.0102	0.0043	0.0193	2	0	I (5)
	${}^{+}S^{*}_{\text{POGA}}$	0.2399	0.6002	0.0015	0.0092	0.0041	0.0030	0	0	I (5)
Yu04	${}^{1}S^{*}_{\text{NSGA-II}}$	0.2565	0.8259	0.0015	0.0122	0.0024	0.0063	0	0	III (8)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.2347	0.6865	0.0014	0.0109	0.0042	0.0114	0	0	II (7)
	${}^{+}S^{*}_{\text{POGA}}$	0.3228	0.3217	0.0042	0.0145	0.0059	0.0229	0	0	I (4)
Yu05	${}^{1}S^{*}_{\text{NSGA-II}}$	0.3940	0.5598	0.0052	0.0248	0.0048	0.0422	0	0	II (8)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.3226	0.4103	0.0043	0.0187	0.0062	0.0405	0	0	II (8)
	${}^{+}S^{*}_{\text{POGA}}$	0.4571	0.3789	0.0037	0.0114	0.0039	0.0106	1	1	I (5)
Yu06	${}^{1}S^{*}_{\text{NSGA-II}}$	0.5054	0.5949	0.0040	0.0171	0.0043	0.0266	0	0	II (7)
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.4470	0.4596	0.0037	0.0141	0.0041	0.0258	1	1	II (7)
	${}^{+}S^{*}_{\text{POGA}}$	0.5798	0.4629	0.0052	0.0153	0.0090	0.0240	0	1	I (5,55)
Yu07	${}^{1}S^{*}_{\text{NSGA-II}}$	0.6682	0.7172	0.0058	0.0225	0.0099	0.0410	1	1	Dominated
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.5746	0.5582	0.0051	0.0188	0.0082	0.0394	0	1	I (5,55)
	${}^{+}S^{*}_{\text{POGA}}$	0.4596	0.2543	0.0067	0.0216	0.0199	0.0702	0	1	II (7)
Yu08	${}^{1}S^{*}_{\text{NSGA-II}}$	0.6830	0.3800	0.0105	0.0303	0.0260	0.0682	2	2	Dominated
	${}^{2}S^{*}_{\text{NSGA-II}}$	0.4578	0.2519	0.0068	0.0206	0.0217	0.0488	0	1	I (5)

On the premise of this analysis, it was concluded that POGA not only calibrated successfully the model BEMUS, but also performed overall better than NSGA-II.

CONCLUSIONS

In this paper the Preference Ordering Genetic Algorithm (POGA), a new MOGA was presented. It was claimed that its ranking scheme would make it suitable to successfully tackle MOP with a massive number of objective functions and we demonstrated it by comparing the scalability features of POGA with a widely accepted algorithm, NSGA-II, on a test function common in the literature.
POGA was hen applied to the automatic calibration of a sewer drainage network model, BEMUS, for the experimental urban catchment Miljakovac (in Belgrade, Serbia). The statistics presented suggest that POGA was able to successfully calibrate the model BEMUS. A comparison with the calibration performed by NSGA-II highlighted that not only was POGA able to find overall better solutions but it also identified a set of best-compromise parameter values that was further analysed

because it represents a logical choice from an engineering point of view. This parameter set proved to be qualitatively superior to the best parameter sets identified by NSGA-II.

FUTURE WORK

A comprehensive investigation on the complexity of the algorithm has not been provided yet. Although empirical results suggest that the ranking procedure should scale less the than exponentially, a detailed analysis is required and it is currently being undertaken by the authors.

The present work claims and proves that the Preference Ordering condition of efficiency of order, denoted throughout as PO_k, is a promising ranking procedure to be embodied in MOGAs to enhance the exploration of the decision variables space. Further research is necessitated to investigate the effects of extending this methodology to the Preference Ordering condition of efficiency of order and degree ($PO_{k,z}$).

Even though POGA draws heavily from NSGA-II, the latter is mainly chosen as a proof of concept to show the potential of Preference Ordering as a ranking procedure embedded in MOGAs. Therefore, it is the author's firm belief that other algorithms might benefit from the concept presented in the paper.

REFERENCES

[1] Branke, J. and Deb, K., (2004). "Integrating User Preferences into Evolutionary Multi-Objective Optimization", Technical Report, Kanpur Genetic Algorithm Laboratory (KanGAL), 2004004.

[2] Branke, J., Kaubler, T., and Schmeck, H., (2001). "Guidance in Evolutionary Multi-Objective Optimization", *Advances in Engineering Software*, vol. 32:499-507.

[3] Coello Coello, C. A., (2000). "Handling Preferences in Evolutionary Multiobjective Optimization: A Survey", in 2000 Congress on Evolutionary Computation, vol. 1 pp. 30-37, Piscataway, New Jersey: IEEE Service Center.

[4] Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic.

[5] Corne, D. W., Knowles, J. D., and Oates, M. J., (2000). "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization", in Proceedings of the Parallel Problem Solving from Nature {VI} Conference,pp. 839-848, M. Schoenauer and K. Deb and G. Rudolph and X. YAO and E. Lutton and J. J. Merelo and H. P. Schwefel Eds. Springer. Lecture Notes in Computer Science No. 1917.

[6] Cvetkovi´c, D. and Parmee, I. C., (2002). "Preferences and their Application in Evolutionary Multiobjective Optimisation", *IEEE Transaction on Evolutionary Computation*, vol. 6(1):42-57.

[7] Das, I. and Dennis, J. E., (1998). "Normal Boundary Intersection A New Method for generating the Pareto Surface in nonlinear multi objective optimization problems", *SIAM J. on Optimization*, vol. 8(3):631-657.

[8] Deb, K., (1999). "Multi-Objective Evolutionary Algorithms: Introducing Bias Among Pareto-Optimal Solutions", Technical Report, Kanpur Genetic Algorithms Laboratory (KanGAL), 99002.

[9] Deb, K., (2001). *Multi-Objective Optimization using Evolutionary Algorithms* John Wiley & Sons, Chichester, UK.

[10] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., (2000). "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", Technical Report, Kanpur Genetic Algorithm Laboratory (KanGAL), 200001.

[11] Deb, K., Thiele, L., and Zitzler, E., (2002). "Scalable Multi-Objective Optimization Test Problems", in IEEE Congress on Evolutionary Computation, (CEC 2002), Piscataway (NJ): IEEE press.
[12] Djordjevic, S., (2001). "A mathematical model of the interaction between surface and buried pipe flow in urban runoff and drainage." Ph.D. Faculty of Civil Engineering, University of Belgrade, Belgrade.
[13] Goldberg, D. E., (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Mass: Addison-Wesley.
[14] Holland, J. H., (1975). *Adaptation in Natural and Artificial Systems* Ann Arbor, MI: The University of Michigan Press.
[15] Horn, J., Nafpliotis, N., and Goldberg, D. E., (1994). "A Niched Pareto Genetic Algorithm for Multiobjective Optimization", in Proceedings of the First {IEEE} Conference on Evolutionary Computation, {IEEE} World Congress on Computational Intelligence, vol. 1 pp. 82-87, Piscataway, New Jersey: IEEE Service Center.
[16] Knowles, J. and Corne, D., (1999). "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation", in Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99), vol. 1 pp. 98-105, P. J. Angeline and Z. Michalewicz and M. Schoenauer and X. Yao and A. Zalzala Eds. IEEE Press.
[17] Madsen, H. and Khu, S. T., (2002). "Parameter estimation in hydrological modelling using multi-objective optimization", in Proceedings of the Fifth International Conference on Hydroinformatics, Software Tools and Management Systems, vol. 2 pp. 1160-1165, I. D. Cluckie and D. Han and J. P. Davis and S. Heslop Eds.
[18] Ponce, V. M. and Yevjevich, V., (1978). "Muskingum-Cunge Method with Variable Parameters", *Journal of the Hydraulics Division*, vol. 104(HY12):1663-1667.
[19] Radojkoviæ, M. and Maksimoviæ, C., (1984). "Development, testing and application of Belgrade urban drainage model", in 3rd ICUD, Goteborg, vol. 4
[20] Srinivas, N. and Deb, K., (1995). "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, vol. 2(3):221-248.
[21] Van Veldhuizen, D. A. and Lamont, G. B., (2000). "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", *Evolutionary Computation*, vol. 8(2):125-147.
[22] Zitzler, E., Laumanns, M., and Thiele, L., (2001). "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", Swiss Federal Institute of Technology (ETH) Zurich, TIK, 103.
[23] Zitzler, E. and Thiele, L., (1999). "Multiobjective Evolutionary Algorithms: A Comparative Case study and the Strength Pareto Approach", *IEEE Transaction on Evolutionary Computation*, vol. 3(4):257-271.
[24] Zitzler, E., Thiele, L., and Deb, K., (2000). "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, vol. 8(2):173-195.