(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 August 2012 (09.08.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/106212 A1

(51) International Patent Classification:
G06N 3/12 (2006.01)

(21) International Application Number:
PCT/US2012/022972

(22) International Filing Date:
27 January 2012 (27.01.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/437,846 31 January 2011 (31.01.2011) US
13/356,391 23 January 2012 (23.01.2012) US

(71) Applicant *(for all designated States except US)*: **THE PENN STATE RESEARCH FOUNDATION** [US/US]; Intellectual Property Office, The Pennsylvania State University, 113 Technology Center, University Park, Pennsylvania 16802 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **HADKA, David** [US/US]; 709 West Cherry Lane #4, State College, Pennsylvania 16803 (US). **REED, Patrick** [US/US]; 160 Waypoint Circle, State College, Pennsylvania 16801 (US).

(74) Agent: **SAHASRABUDDHE, Laxman**; Park, Vaughan, Fleming & Dowler LLP, 2820 Fifth Street, Davis, California 95618 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: BORG: AN AUTO-ADAPTIVE MANY - OBJECTIVE EVOLUTIONARY COMPUTING FRAMEWORK

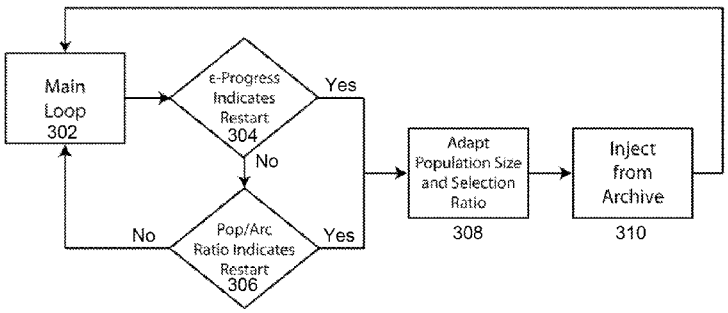

FIG. 3

(57) **Abstract**: Methods and apparatuses for performing evolutionary based optimization are described. Specifically, some embodiments feature: a dominance archive, auto-adaptive operators, detection of search stagnation, exploitation of randomized restarts to escape local optima, and/or selection of recombination operators based on their success in generating high quality solutions.

WO 2012/106212 A1

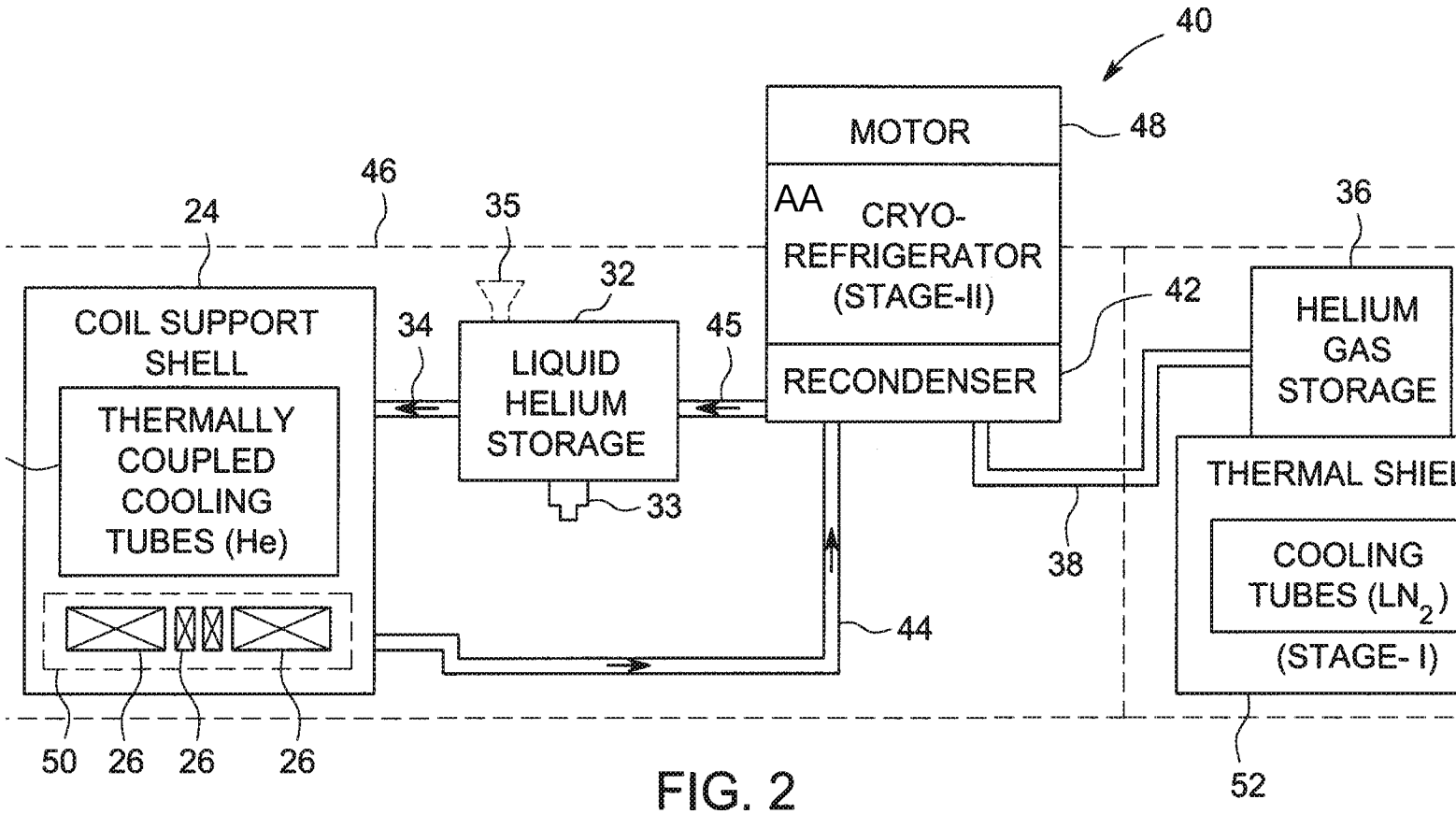

FIG. 2

22 TUBES DE REFROIDISSEMENT COUPLES THERMIQUEMENT (He)
24 COQUE SUPPORT DE BOBINES
32 STOCKAGE D'HELIUM LIQUIDE
36 STOCKAGE D'HELIUM LIQUIDE
42 RECONDENSEUR
AA CRYOREFRIGERATEUR (ETAGE II)
48 MOTEUR
52 BOUCLIER THERMIQUE
54 TUBES DE REFROIDISSEMENT (LN2)
BB (ETAGE-1)

[0005] Further, some studies have observed that the selection mechanism promoting diverse solutions along the entire extent of the tradeoff surface can cause deterioration. Deterioration occurs whenever the solution set discovered by an ECBOT at time i contains one or more solutions dominated by a solution discovered at some earlier point in time $j < i$. In the extreme, deterioration can cause an ECBOT to diverge away from the Pareto front. Many state-of-the-art ECBOTs in use today are not capable of avoiding deterioration.

[0006] Recently, studies have attempted to rectify these issues by improving various components of ECBOTs. One such approach attempts to eliminate dominance resistance by utilizing more stringent dominance relations. These studies tend to focus on the "knee" region of the Pareto front and fail to produce solutions along the entire extent of the tradeoff surface.

SUMMARY

[0007] Some embodiments described herein feature methods and apparatuses for performing evolutionary computing based optimization. Specifically, some embodiments include one or more of the following features: an ε-dominance archive, auto-adaptive operators, detection of search stagnation, exploitation of randomized restarts to escape local optima, and/or selection of recombination operators based on their success in generating high quality solutions.

[0008] Some embodiments described herein determine an archive of solution vectors for an optimization problem. Specifically, a new solution vector for the optimization problem can be determined based on a population of solution vectors and the archive of solution vectors. In response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors, the one or more solution vectors can be replaced with the new solution vector in the archive of solution vectors. In response to determining that ε-progress has not occurred, the population based on the archive of solution vectors can be recreated. In some embodiments, determining that ε-progress has not occurred can involve comparing a current value of a ε-progress counter with a previously stored value of the ε-progress counter and/or determining that a ε-progress timer has expired.

[0009] According to one definition, for a given ε value, new solution vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ results in ε-progress if and only if, for all solution vectors $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ in the archive of solution vectors, $\left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor \neq \left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor$ and $\left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor$ does not Pareto dominate $\left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor$.

[0010] In some embodiments, the new solution vector dominates an existing solution vector in the archive of solution vectors if and only if the new solution vector ε-dominates the existing solution vector. According to one definition, for a given $\varepsilon > 0$ value, a solution vector $\mathbf{u}$

$= (u_1, u_2, \ldots, u_M)$ ε-dominates another solution vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if $\forall i \in \{1,2,...,M\}, u_i \leq v_i + \varepsilon$ and $\exists j \in \{1,2,...M\}, u_j < v_j + \varepsilon$.

[0011] In some embodiments, the new solution vector dominates an existing solution vector in the archive of solution vectors if and only if the new solution vector ε-box dominates the existing solution vector. According to one definition, for a given $\varepsilon > 0$ value, solution vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ ε-box dominates solution vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if: $\left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor \prec \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor$, or $\left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor = \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor$ and $\left\| \mathbf{u} - \varepsilon \left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor \right\| < \left\| \mathbf{v} - \varepsilon \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor \right\|$.

[0012] In some embodiments, a ratio between a size of the population and a size of the archive of solution vectors can be determined. In response to determining that a difference between the determined ratio and a target ratio is greater than a pre-determined threshold, the population can be recreated based on the archive of solution vectors. In some embodiments, recreating the population can involve: adjusting a size of the population so that a ratio between the size of the population and the size of the archive of solution vectors is substantially equal to a target ratio, and adjusting a tournament size so that a selection pressure of a tournament selection process that is used for recreating the population is substantially equal to a target selection pressure.

[0013] In some embodiments, recreating the population based on the archive of solution vectors can include: emptying the population; adding solution vectors from the archive of solution vectors to the population; randomly selecting a solution vector from the archive of solution vectors, and one or more solution vectors from the population; recombining the randomly selected solution vectors to obtain a mutated solution vector; adding the mutated solution vector to the population; and if the mutated solution vector dominates one or more solution vectors in the archive of solution vectors, replacing the one or more solution vectors in the archive of solution vectors by the mutated solution vector.

[0014] In some embodiments, determining the new solution vector for the optimization problem can include: selecting a recombination operator from a set of recombination operators based on a set of probability values, wherein each recombination operator corresponding to a probability value; determining the new solution vector using the selected recombination operator; and in response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors, increasing the probability value associated with the selected recombination operator.

[0015] The set of recombination operators can include one or more of: a simulated binary crossover operator; a differential evolution operator; a parent-centric crossover operator; a unimodal normal distribution operator; a simplex crossover operator; or a uniform mutation operator.

BRIEF DESCRIPTION OF THE FIGURES

[0016] FIG. 1 illustrates a process for updating an ε-box dominance archive in accordance with some embodiments described herein.

[0017] FIG. 2 illustrates how ε-progress can be measured in accordance with some embodiments described herein.

[0018] FIG. 3 presents a flowchart that illustrates a process for restarting the optimization in accordance with some embodiments described herein.

[0019] FIG. 4 illustrates a process for creating a new population during restart in accordance with some embodiments described herein.

[0020] FIGs. 5A-5F show the offspring distribution generated by different operators in accordance with some embodiments described herein.

[0021] FIG. 6 presents a flowchart that illustrates a process for performing a multi-objective optimization process in accordance with some embodiments described herein.

[0022] FIG. 7 illustrates a computer system in accordance with some embodiments described herein.

DETAILED DESCRIPTION

[0023] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0024] An optimization problem with M objectives can be defined as:

$$\begin{aligned} \underset{\mathbf{x}\in\Omega}{\text{minimize}} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0, \ \forall i \in \mathcal{E}, \\ & c_j(\mathbf{x}) \leq 0, \ \forall j \in \mathcal{I}. \end{aligned}$$

[0025] We call $\mathbf{x}$ the decision variables and Ω the decision space. The decision variables can have real or complex values. Without loss of generality, let $\mathbf{x} = (x_1, x_1, \ldots, x_L)$, $x_i \in \Re$, of fixed length L. The sets $\mathcal{E}$ and $\mathcal{I}$ contain the indices for all equality and inequality constraints, respectively. The feasible region is the set of all decision variables in Ω that satisfy all constraints. The optimization problem can be constrained or unconstrained (i.e., when sets $\mathcal{E}$ and $\mathcal{I}$ are null). Some embodiments described herein can be used for problems with $M \geq 4$ objectives.

[0026] Embodiments disclosed herein can be used for *a priori* optimization or *a posteriori* optimization. According to one definition, *a priori* optimization refers to an optimization process in which the weights and/or preferences of the different components of the cost function are known *a priori*. According to one definition, *a posteriori* optimization refers to an optimization process in which the weights and/or preferences of the different components of the cost function are not known *a priori*. When used for *a posteriori* optimization, some embodiments described herein can generate a set of potential solutions allowing a decision maker to explore the various tradeoffs and identify the preferred solution(s).

[0027] The notion of optimality when tradeoffs exist between solutions can be captured by Pareto dominance and the Pareto optimal set. The terms "Pareto dominance," "Pareto optimal set," and "Pareto front" can be defined as follows.

[0028] A vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ Pareto dominates another vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if $\forall i \in \{1,2,...,M\}, u_i \leq v_i$ and $\exists j \in \{1,2,...M\}, u_j < v_j$. Pareto dominance is denoted by $\mathbf{u} \prec \mathbf{v}$.

[0029] For a given multi-objective problem, the Pareto optimal set is defined by

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

[0030] For a given multi-objective problem with Pareto optimal set $\mathcal{P}^*$, the Pareto front is defined by

$$\mathcal{PF}^* = \{F(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^*\}$$

[0031] In ECBOTs, the Pareto dominance relation is applied to the objectives. For convenience, we use $\mathbf{x} \prec \mathbf{y}$ interchangeably with $F(\mathbf{x}) \prec F(\mathbf{y})$. Two solutions are non-dominated if neither Pareto dominates the other. Using this terminology, the goal of *a posteriori* optimization is the enumeration or approximation of the Pareto front.

[0032] Some embodiments described herein present an ECBOT that is specifically designed for handling multi-objective (especially when the number of objective is greater than or equal to four), multimodal problems. Specifically, some embodiments described herein include

one or more of the following features: (1) an ε-box dominance archive for maintaining convergence and diversity throughout search; (2) ε-progress, which is a computationally efficient measure of search progression/stagnation; (3) an adaptive population sizing operator based on a use of time continuation to maintain search diversity and to facilitate escape from local optima; (4) multiple recombination operators to enhance search in a wide assortment of problem domains; and (5) a steady-state, elitist model of ε-ECBOT, which can be easily extended for use on parallel architectures.

ε-box dominance

[0033] For a given $\varepsilon > 0$, a vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ ε-dominates another vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if $\forall i \in \{1,2,...,M\}, u_i \leq v_i + \varepsilon$ and $\exists j \in \{1,2,...M\}, u_j < v_j + \varepsilon$.

[0034] In addition to the theoretical benefits of guaranteed convergence and diversity, ε-dominance provides a minimum resolution which effectively bounds the archive size. This is of practical importance to decision makers, who are able to define ε using domain-specific knowledge of their precision goals or computational limits. In practice, it is useful to specify different ε values for each objective; however, without loss of generality, a single ε value is used in the following discussion for the sake of clarity and ease of discourse.

[0035] ε-box dominance is a variation of ε-dominance, and is defined as follows. For a given $\varepsilon > 0$, a vector $\mathbf{u} = (u_1, u_2, \ldots, u_M)$ ε-box dominates another vector $\mathbf{v} = (v_1, v_2, \ldots, v_M)$ if and only if one of the following is true: (1) $\left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor \prec \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor$, or (2) $\left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor = \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor$ and $\left\| \mathbf{u} - \varepsilon \left\lfloor \frac{\mathbf{u}}{\varepsilon} \right\rfloor \right\| < \left\| \mathbf{v} - \varepsilon \left\lfloor \frac{\mathbf{v}}{\varepsilon} \right\rfloor \right\|$.

[0036] ε-box dominance is denoted by $\mathbf{u} \prec_\varepsilon \mathbf{v}$. Conceptually, the ε-box dominance archive divides the objective space into hyperboxes with side-length ε, called ε-boxes. The ε-box in which a solution vector resides is determined using the ε-box index vector. We use the notation

$$\left\lfloor \frac{\mathbf{u}}{\epsilon} \right\rfloor = \left(\left\lfloor \frac{u_1}{\epsilon} \right\rfloor, \left\lfloor \frac{u_2}{\epsilon} \right\rfloor, \ldots, \left\lfloor \frac{u_M}{\epsilon} \right\rfloor \right)$$

for computing the ε-box index vector, where $\lfloor . \rfloor$ is the floor function. Note that ε-box dominance is determined using this index vector rather than the object values. Case (2) in the ε-box dominance definition shown above covers the situation in which two or more solutions reside in the same ε-box. In this situation, the solution nearest the lower-left (minimized) corner of the ε-box dominates any other solutions in the same ε-box.

[0037] FIG. 1 illustrates a process for updating an ε-box dominance archive in accordance with some embodiments described herein.

[0038] The input to the process is the new solution vector **x** that is being added to the archive. The process outputs a "true" value if solution vector **x** is added to the archive; a "false" value otherwise. If vector **x** ε-box dominates one or more vectors in the archive, then the process removes all such vectors (operation 102), and adds vector **x** to the archive (operation 104). On the other hand, if a vector in the archive ε-box dominates **x**, then the process returns a "false" value without changing the archive (operation 106). In some embodiments, the archive of solution vectors is an enumeration or an approximation of the Pareto front.

[0039] The pseudocode shown in FIG. 1 has been presented only for purposes of illustration and description. It is not intended to limit the present invention to the form disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. For example, a process for updating an ε-dominance archive can be obtained by replacing the ε-box dominance operator "$\prec_\varepsilon$" with an ε-dominance operator in FIG. 1.

ε-progress

[0040] While the ε-box dominance archive guarantees convergence and diversity, this guarantee is subject to the solutions produced by the ECBOT. ECBOTs tend to fail on multimodal problems due to pre-convergence to local optima causing search to stagnate. Some embodiments described herein use a computationally efficient extension to the ε-box dominance archive for measuring search progression called ε-progress. The inability of an ECBOT to maintain ε-progress indicates search stagnation, which can subsequently trigger routines for reviving search.

[0041] ε-progress occurs when a solution **x** passed to the process shown in FIG. 1 is accepted into the archive such that no existing member of the archive existed with the same ε-box index vector.

[0042] FIG. 2 illustrates how ε-progress can be measured in accordance with some embodiments described herein.

[0043] In the 2-dimensional example shown in FIG. 2, existing archive members are indicated by a "•," and the ε-boxes dominated by these members are shaded. New solutions that are added to the archive are indicated by a "×". Solutions 202 and 204 depict occurrences of ε-progress. The new solutions reside in previously unoccupied ε-boxes. Solution 206 shows the situation in which the new solution is accepted into the archive, but since it resides in an

occupied ε-box it does not count towards ε-progress because the improvement is below the threshold ε.

[0044] ε-progress supplements the use of ε as the problem resolution by mandating ε as the minimum threshold for improvement. In some embodiments described herein, the process must periodically produce at least one solution whose improvement exceeds this threshold to avoid stagnation. If stagnation is detected, appropriate action can be taken to either revive search or terminate the process.

[0045] In some embodiments described herein, the ε-box dominance archive increments a counter every time ε-progress occurs. This counter can be periodically checked after a user-specified number of evaluations or after a timer expires. If the counter is unchanged from the previous check, then the ECBOT failed to produce significant improvements and a restart mechanism can be triggered.

Restarts

[0046] Some embodiments described herein use multiple-epoch runs. Some multiple-epoch runs are characterized by periodically emptying the population, retaining the best solution(s), and repopulating with new randomly-generated solutions.

[0047] Between two epochs, the optimization process can perform a restart. Restarts are a mechanism for reviving search after stagnation is detected using ε-progress. In some embodiments described herein, a restart can include three functions: (1) the search population size is adapted to remain proportional to the archive size; (2) the selection operator is adapted to maintain a constant selection pressure; and (3) a new population is created by injecting from the archive, i.e., the population is emptied and repopulated with solutions from the archive, with any remaining slots filled by mutated archive solutions.

[0048] FIG. 3 presents a flowchart that illustrates a process for restarting the optimization in accordance with some embodiments described herein.

[0049] After a certain number of evaluations or after a timer expires, the ECBOT can break out of main loop 302 to check if ε-progress indicates a restart should be performed (operation 304) or the population-to-archive ratio indicates a restart should be performed (operation 306). If either of these two checks (i.e., operations 304 or 306) indicates that a restart should be performed, the population size and selection ratio can be adapted (operation 308), and a new population can be created by injecting from the archive (operation 310). Specifically, the population can be resized and filled with solutions selected from the archive and mutated using uniform mutation applied with probability $1/L$, where L is the number of decision variables. In

addition, the tournament selection size can be adjusted given the new population size to maintain constant selection pressure. Finally, main loop 302 can be resumed.

[0050] FIG. 4 illustrates a process for creating a new population during restart in accordance with some embodiments described herein. The pseudocode shown in FIG. 4 has been presented only for purposes of illustration and description. It is not intended to limit the present invention to the form disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art.

[0051] Once the old members in the population have been removed, the population can be filled with solutions in the archive (operation 402). Next, a new size for the population can be determined (operation 404). The remaining members of the population can then be filled in by mutating randomly chosen members in the archive, and updating the archive with the mutated solution (operation 406). Finally, the tournament size can be set to maintain constant selection pressure (operation 408).

[0052] It has been observed that maintaining a population size proportional to the archive size helps escape local optima on highly multimodal real-world problems. This mechanism of adapting the population size can be built into the optimization process via the use of the injection rate parameter γ. The injection rate specifies the ratio of the population size to the archive size, i.e.,

$$\gamma = \frac{\text{population size}}{\text{archive size}}$$

[0053] In some embodiments described herein, the population-to-archive ratio is maintained throughout the optimization process. At any point during the execution of the optimization process, if the population-to-archive ratio differs from the target γ by more than a predetermined amount (e.g., 25%), the population size is adapted. This strategy ensures the population size remains commensurate with the Pareto front discovered by the ECBOT. By using the archive size as a proxy for problem difficulty, we assume the population should grow proportionally with problem difficulty.

[0054] While adaptive population sizing can aid an ECBOT by maintaining a large set of diverse solutions, an optimization process with dynamically increasing search populations can be prone to genetic drift stall failures due to weakening selection pressure. Therefore, in some embodiments described herein, the selection pressure is kept substantially constant during the optimization process.

[0055] The following selection pressure analysis is based on tournament selection, which is the selection mechanism used in some ECBOTs described in this disclosure. The following analysis is for illustration purposes only, and is not intended to limit the present invention to the

form disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. The probability of selecting the i-th best individual from a population of size λ using a tournament size q can be expressed as:

$$p_i = \lambda^{-q}((\lambda - i + 1)^q - (\lambda - i)^q).$$

[0056] Given the population size λ, we can compute the value of q using the above equation to maintain any desired probability of selection p_i (note that the probability of selection p_i corresponds to the selection pressure). In some embodiments, the selection pressure may remain equal to binary tournament selection with a population size of 100. In other words, in these embodiments, the selection ratio is $\tau = 0.2$. The tournament size q can be computed using the above equation, or it can be approximately computed as

$$\text{tournament size} = \max(2, \lfloor \tau\lambda \rfloor).$$

Auto-adaptive multi-operator recombination

[0057] One of the problems encountered when using ECBOTs is the inability to know *a priori* which recombination operator performs best on a given problem. Some embodiments described herein establish a feedback loop in which operators that produce more successful offspring are rewarded by increasing the number of offspring produced by that operator. Specifically, given $K > 1$ operators, we maintain the probabilities $\{Q_1, Q_2, \ldots, Q_K\}$ of applying each operator to produce the next offspring. In some embodiments described herein, these probabilities can be initialized to $Q_i = 1/K$. The operator that is applied in an iteration of the optimization process is selected based on these probability values. Periodically, these probabilities are updated by first counting the number of solutions in the ε-box dominance archive that were produced by each operator, $\{C_1, C_2, \ldots, C_K\}$, and updating each Q_i so that operators that tend to produce high quality solutions have a higher probability of being selected. Specifically, in some embodiments, each Q_i can be updated using the following expression:

$$Q_i = \frac{C_i + \varsigma}{\sum_{j=1}^{K}(C_j + \varsigma)}.$$

The constant $\varsigma > 0$ prevents the operator probabilities from reaching 0, thus ensuring no operators are "lost" during the execution of the process. In some embodiments, $\varsigma = 1$. Other expressions for updating the Q_i values will be apparent to practitioners skilled in the art.

[0058] In some embodiments described herein, the feedback loop updates the probabilities by counting the number of solutions produced by each operator in the ε-box dominance archive. Since the ε-box dominance archive maintains the best solutions in terms of

both convergence and diversity, selecting operators based on the Q_i values favors operators that produce offspring with these qualities.

[0059] Some embodiments described herein can be viewed as a class of evolutionary optimization approaches whose operators are adaptively selected based on the problem and the decision variable encoding. Specifically, some embodiments use the following parent-centric, mean-centric, uniformly distributed and self-adaptive real-valued operators: (1) Simulated Binary Crossover (SBX), (2) Differential Evolution (DE), (3) Parent-Centric Crossover (PCX), (4) Unimodal Normal Distribution Crossover (UNDX), (5) Simplex Crossover (SPX), and (6) Uniform Mutation (UM) applied with probability $1/L$. In some embodiments, the offspring produced by SBX, DE, PCX, UNDX and SPX are mutated using Polynomial Mutation (PM).

[0060] FIGs. 5A-5F show the offspring distribution generated by different operators in accordance with some embodiments described herein.

[0061] In these figures, parents are indicated by a "•." The arrows shown in FIG. 5B correspond to the difference vectors. As shown in these figures, some operators tend to generate solutions along a single axis, which can degrade their efficacy on problems with high degrees of conditional dependencies among its decision variables. Other operators, on the other hand, do not exhibit this tendency.

[0062] Some embodiments described herein select the recombination operator based on the Q_i probability values which correspond to how often the operator produces successful offspring. For example, from the six operators that were described above, it is expected that some embodiments described herein will automatically adapt to use operators that do not tend to generate solutions along a single axis for rotated, epistatic problems which have high degrees of conditional dependencies among its decision variables.

[0063] FIG. 6 presents a flowchart that illustrates a process for performing a multi-objective optimization process in accordance with some embodiments described herein.

[0064] In the optimization process, a recombination operator can be selected using the above-described adaptive selection process (block 604). If the recombination operator requires k parents, one parent can be selected uniformly at random from archive 608. Archive 608 can be an ε-dominance archive or an ε-box dominance archive. The remaining $k-1$ parents can be selected from population 606 using tournament selection. The resulting offspring can be evaluated (block 602) and considered for inclusion in population 606 and archive 608.

[0065] In some embodiments, if the offspring ε-dominates or ε-box dominates one or more population members, the offspring replaces one of these dominated members randomly. If the offspring is dominated by at least one population member, the offspring is not added to the

population. Otherwise, the offspring is non-dominated and replaces a randomly-selected member of the population. Inclusion in the archive can be performed using an archive update process, e.g., the one that was described in reference to FIG. 1.

[0066] In each iteration of the iterative optimization process shown in FIG. 6, the optimization process can produce one offspring. After a certain number of iterations have been performed and/or after a certain amount of time has elapsed, ε-progress and the population-to-archive ratio can be checked as described in reference to FIG. 3.

[0067] If a restart is required, the iterative process shown in FIG. 6 halts and the restart process is invoked as described in reference to FIG. 3. Note that the iterative process shown in FIG. 6 corresponds to main loop 302 in FIG. 3. Once the restart has completed, the iterative process shown in FIG. 6 is resumed and this process repeats until termination.

[0068] In some embodiments the process may terminate if: (1) a predetermined number of iterations of the optimization process have been performed, (2) a certain amount of time has elapsed, (3) if ε-progress has not occurred after a certain number of iterations have been performed, and/or (4) if ε-progress has not occurred after a certain amount of time has elapsed. These termination conditions were presented only for purposes of illustration and description. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art.

[0069] Once the optimization process terminates, the archive of solution vectors can be provided as an output of the optimization process. Note that the archive of solution vectors can be viewed as an enumeration or approximation of the Pareto front for the optimization problem.

[0070] Some embodiments described herein scale linearly with the archive size. If we let selection, recombination and dominance checks take constant time and assume γ is a constant, then some embodiments have a runtime complexity of $O(\eta A)$, where η is the number of function evaluations and A is the archive size. As explained above, the archive size is controlled by the value of ε, which can be chosen by decision makers based on either domain-specific knowledge of their precision goals or computational limits. By scaling ε, an ECBOT described herein can be made to run more efficiently at the cost of producing more approximate solutions.

[0071] Exploring the limit behavior of an optimization process as the runtime goes to infinity, is important from a theoretical view. It is not necessary for an optimization process to have guaranteed convergence to be practically useful, but issues like pre-convergence and deterioration that arise in some optimization processes make such results informative. In fact, most ECBOTs do not have guaranteed convergence. The main crux of such convergence proofs is the assumption that there exists a nonzero probability of generating Pareto optimal solutions.

[0072] According to some studies, two conditions are necessary for convergence. The first necessary condition is that the recombination operators should have diagonal-positive transition matrices. In some embodiments described herein the tournament selection operates with replacement and the recombination operators have a form of mutation in which the entire decision space is reachable. Therefore, in these embodiments, the conditions for diagonal-positive transition matrices are satisfied.

[0073] The second necessary condition for guaranteed convergence is elite preservation. It has been shown that the ε-dominance archive satisfies elite preservation because an existing solution in the ε-dominance archive is replaced by a new solution only if the new solution ε-dominates the existing solution. The ε-box dominance archive that is used by some embodiments described herein also satisfies elite preservation for similar reasons. In addition to satisfying elite preservation, it has also been shown that the ε-box dominance archive preserves the diversity of solutions.

[0074] Therefore, the sequence of solutions generated by some embodiments described herein converges completely and in the mean to the set of minimal elements (the Pareto optimal set) as time goes to infinity.

[0075] FIG. 7 illustrates a computer system in accordance with some embodiments described herein.

[0076] A computer system can generally be any system that can perform computations. Specifically, a computer system can be a microprocessor, a network processor, a portable computing device, a personal organizer, a device controller, a distributed or cloud computing environment, or a computational engine within an appliance, or any other computing system now known or later developed. Computer system 702 comprises processor 704, memory 706, and storage 708. Computer system 702 can be coupled with display 714, keyboard 710, and pointing device 712. Storage 708 can generally be any device that can store data. Specifically, a storage device can be a magnetic, an optical, or a magneto-optical storage device, or it can be based on flash memory and/or battery-backed up memory. Storage 708 can store application 716, operating system 718, and data 720.

[0077] Application 716 can include instructions that when executed by computer 702 cause computer 702 to perform one or more processes explicitly or implicitly described in this disclosure.

CONCLUSION

[0078] The above description is presented to enable any person skilled in the art to make and use the embodiments. Various modifications to the disclosed embodiments will be readily

apparent to those skilled in the art, and the general principles defined herein are applicable to other embodiments and applications without departing from the spirit and scope of the present disclosure. Thus, the present invention is not limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0079] The data structures and code described in this disclosure can be partially or fully stored on a non-transitory computer-readable storage medium and/or a hardware module and/or hardware apparatus. The term "non-transitory computer-readable storage medium" as used in this disclosure includes all storage mediums with the sole exception of a propagating electromagnetic wave or signal. Specifically, a non-transitory computer-readable storage medium includes, but is not limited to, volatile memory, non-volatile memory, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs), DVDs (digital versatile discs or digital video discs), or other non-transitory media, now known or later developed, that are capable of storing code and/or data.

[0080] Hardware modules or apparatuses described in this disclosure include, but are not limited to, application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), dedicated or shared processors, and/or other hardware modules or apparatuses now known or later developed.

[0081] The methods and processes described in this disclosure can be partially or fully embodied as code and/or data stored in a non-transitory computer-readable storage medium, so that when a computer system reads and executes the code and/or data, the computer system performs the associated methods and processes. The methods and processes can also be partially or fully embodied in hardware modules or apparatuses. Note that the methods and processes can be embodied using a combination of code, data, and hardware modules or apparatuses.

[0082] The foregoing descriptions of embodiments of the present invention have been presented only for purposes of illustration and description. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. The scope of the present invention is defined by the appended claims.

What Is Claimed Is:

1. A method for determining an archive of solution vectors for an optimization problem, the method comprising:

determining a new solution vector for the optimization problem based on a population of solution vectors and the archive of solution vectors;

in response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors, replacing the one or more solution vectors with the new solution vector in the archive of solution vectors; and

in response to determining that ε-progress has not occurred, recreating the population based on the archive of solution vectors.

2. The method of claim 1, further comprising:

determining a ratio between a size of the population and a size of the archive of solution vectors; and

in response to determining that a difference between the determined ratio and a target ratio is greater than a pre-determined threshold, recreating the population based on the archive of solution vectors.

3. The method of any of claims 1 to 2, wherein recreating the population based on the archive of solution vectors includes:

adjusting a size of the population so that a ratio between the size of the population and the size of the archive of solution vectors is substantially equal to a target ratio; and

adjusting a tournament size so that a selection pressure of a tournament selection process that is used for recreating the population is substantially equal to a target selection pressure.

4. The method of any of claims 1 to 3, wherein recreating the population based on the archive of solution vectors includes:

emptying the population;

adding solution vectors from the archive of solution vectors to the population;

randomly selecting a solution vector from the archive of solution vectors, and one or more solution vectors from the population;

recombining the randomly selected solution vectors to obtain a mutated solution vector; and

adding the mutated solution vector to the population.

5. The method of any of claims 1 to 4, wherein determining the new solution vector for the optimization problem includes:

selecting a recombination operator from a set of recombination operators based on a set of probability values, wherein each recombination operator corresponding to a probability value;

determining the new solution vector using the selected recombination operator; and

in response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors, increasing the probability value associated with the selected recombination operator.

6. The method of claim 5, wherein the set of recombination operators includes one or more of:

a simulated binary crossover operator;

a differential evolution operator;

a parent-centric crossover operator;

a unimodal normal distribution operator;

a simplex crossover operator; or

a uniform mutation operator.

7. The method of any of claims 1 to 6, wherein determining that ε-progress has not occurred involves comparing a current value of a ε-progress counter with a previously stored value of the ε-progress counter.

8. The method of any of claims 1 to 7, wherein determining that ε-progress has not occurred involves determining that an ε-progress timer has expired.

9. The method of any of claims 1 to 8, wherein the new solution vector dominates an existing solution vector in the archive of solution vectors if and only if the new solution vector ε-dominates the existing solution vector.

10. The method of any of claims 1 to 9, wherein the new solution vector dominates an existing solution vector in the archive of solution vectors if and only if the new solution vector ε-box dominates the existing solution vector.

11. An apparatus, comprising:

first determining means for determining a new solution vector for the optimization problem based on a population of solution vectors and the archive of solution vectors;

replacing means for replacing the one or more solution vectors with the new solution vector in the archive of solution vectors in response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors; and

first recreating means for recreating the population based on the archive of solution vectors in response to determining that ε-progress has not occurred.

12. The apparatus of claim 11, further comprising:

second determining means for determining a ratio between a size of the population and a size of the archive of solution vectors; and

second recreating means for recreating the population based on the archive of solution vectors in response to determining that a difference between the determined ratio and a target ratio is greater than a pre-determined threshold.

13. The apparatus of any of claims 11 to 12, wherein the first determining means comprises:

selecting means for selecting a recombination operator from a set of recombination operators based on a set of probability values, wherein each recombination operator corresponding to a probability value;

third determining means for determining the new solution vector using the selected recombination operator; and

increasing means for increasing the probability value associated with the selected recombination operator in response to determining that the new solution vector dominates one or more solution vectors in the archive of solution vectors.

14. The apparatus of claim 13, wherein the set of recombination operators includes one or more of:

a simulated binary crossover operator;

a differential evolution operator;

a parent-centric crossover operator;

a unimodal normal distribution operator;

a simplex crossover operator; or

a uniform mutation operator.

15. A computer program for implementing the method of any of claims 1 to 10.

Input: The new solution x being added to the archive.
Output: true if x is added to the archive; false otherwise.

foreach *solution* y *in the archive* **do**
  if x $\prec_\epsilon$ y **then**
    remove y from the archive; ← 102
  else if y $\prec_\epsilon$ x **then**
    return false; ← 106
add x to the archive; ← 104
return true;

FIG. 1

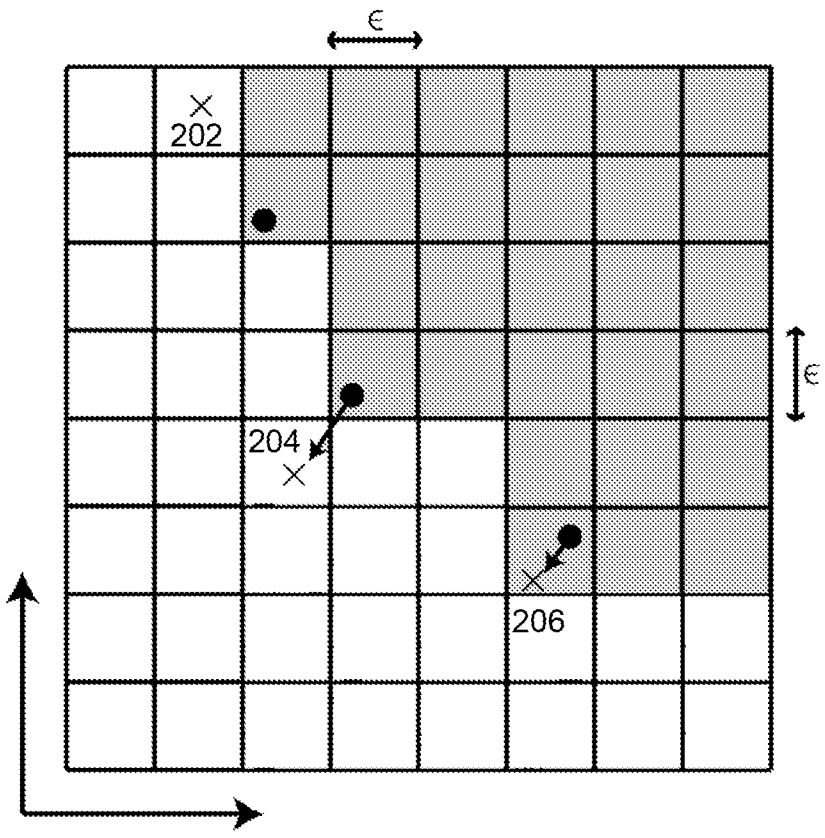

FIG. 2

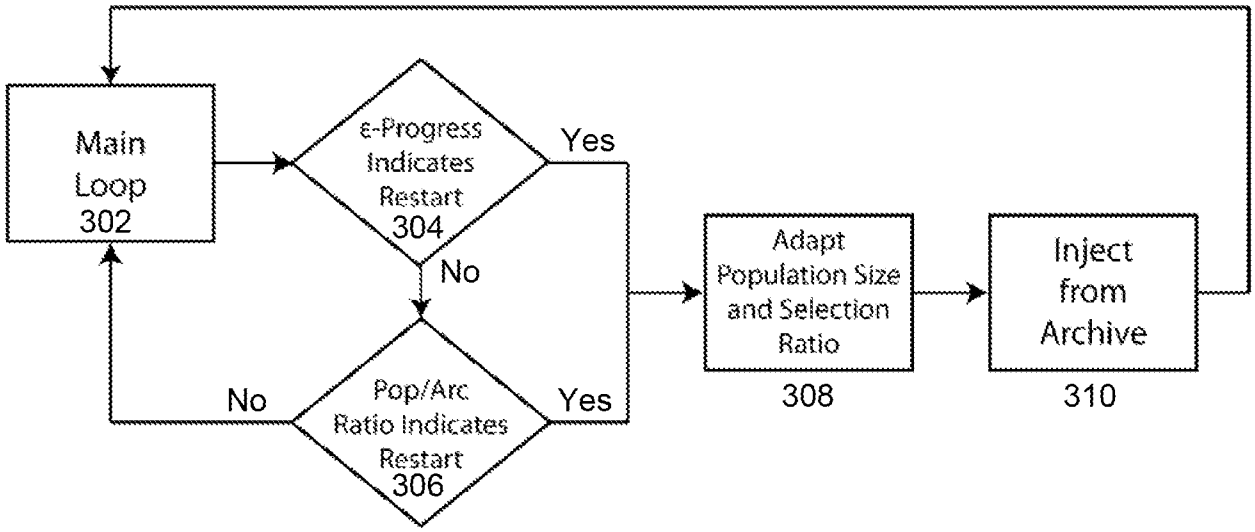

FIG. 3

Input: The current archive, the population-to-archive ratio γ and the selection ratio τ

Output: The population after random restart

Empty the population;
Fill population with all solutions in the archive; ← 402

```
// Compute the size of the new population
```
new_size ← γ * size(archive); ← 404

```
// Inject mutated archive members into the new population
```
while size(population) < new_size **do** ← 406
    new_solution ← select randomly one solution from archive;
    Mutate new_solution using uniform mutation applied with probability $1/L$;
    Add new_solution to population;
    Update archive with new_solution;

```
// Maintain constant selection pressure
```
Set the tournament size to max(2, floor(τ * new_size)); ← 408

FIG. 4

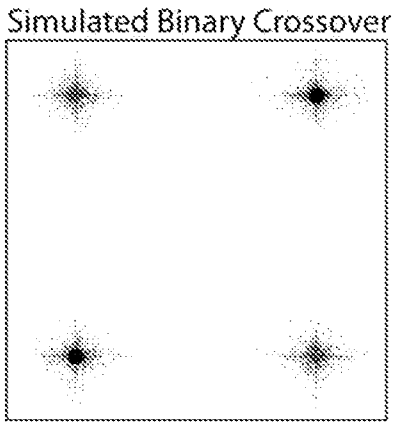

FIG. 5A

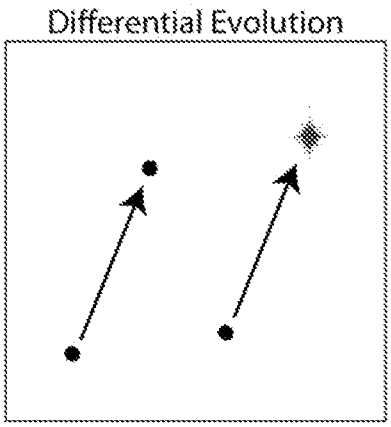

FIG. 5B

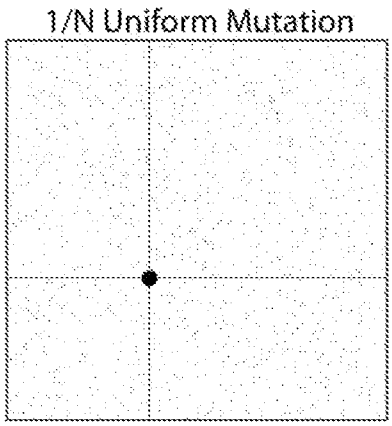

FIG. 5C

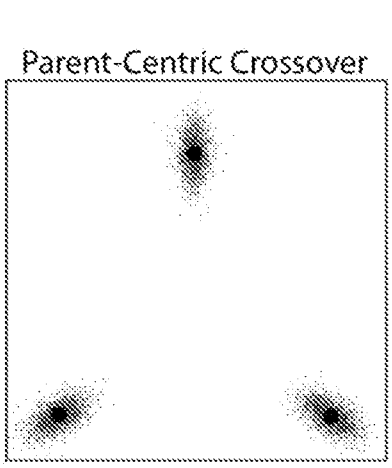

FIG. 5D

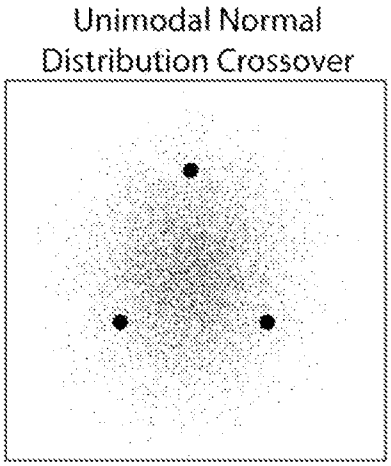

FIG. 5E

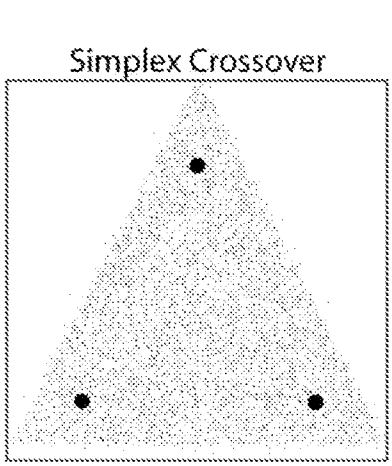

FIG. 5F

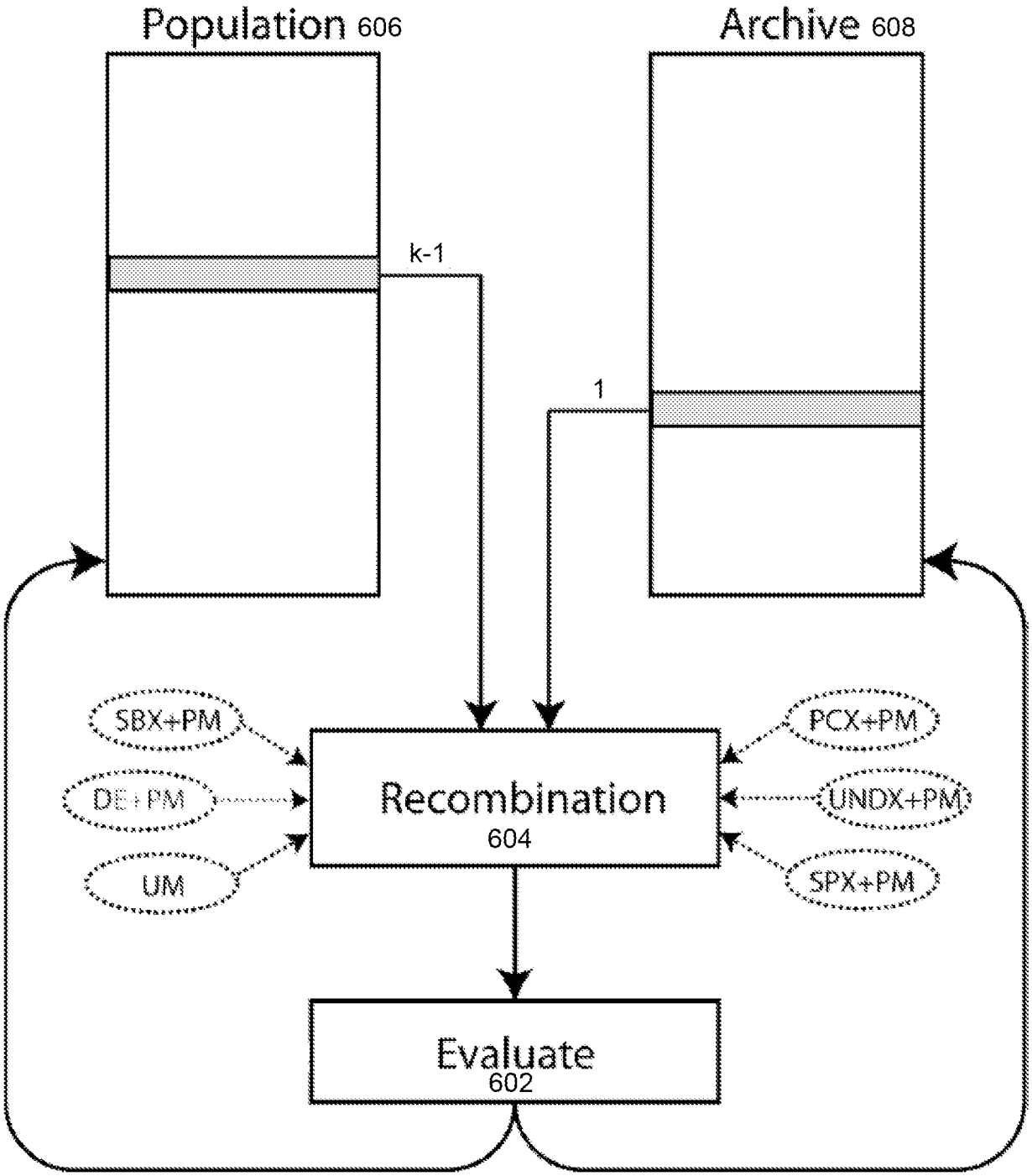
Population 606
Archive 608
k-1
1
SBX+PM
DE+PM
UM
Recombination
604
PCX+PM
UNDX+PM
SPX+PM
Evaluate
602

FIG. 6

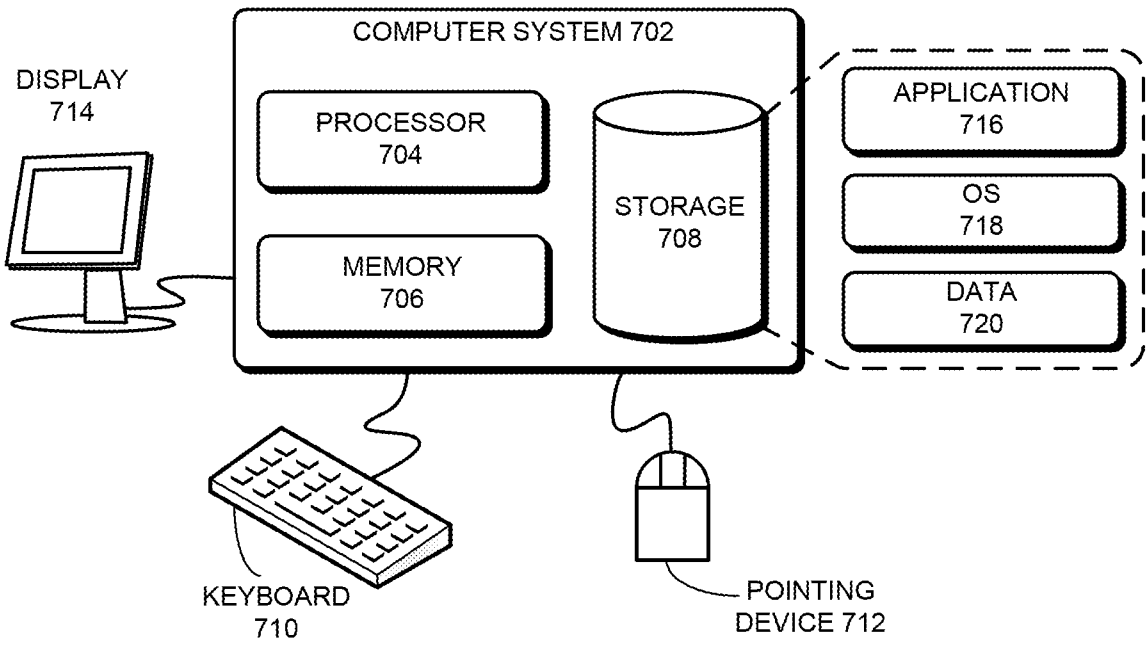
COMPUTER SYSTEM 702
DISPLAY
714
PROCESSOR
704
MEMORY
706
STORAGE
708
APPLICATION
716
OS
718
DATA
720
KEYBOARD
710
POINTING
DEVICE 712

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/022972

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N3/12
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	D. RAAD, A. SINSKE, J. VAN VUUREN: "Robust multi-objective optimization for water distribution system design using a meta-metaheuristic", INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH, vol. 16, no. 5, 24 July 2009 (2009-07-24), pages 595-626, XP55021662, DOI: 10.1111/j.1475-3995.2009.00705.x sections 6 & 9	1-15

-/--

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 March 2012	27/03/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Douarche, Nicolas

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/022972

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	J. NICKLOW, P. REED, D. SAVIC, T. DESSALEGNE, L. HARRELL, A. CHAN-HILTON, M. KARAMOUZ, B. MINSKER, A. OSTFELD, A. SINGH,E. ZECHMAN: "State of the art for genetic algorithms and beyond in water resources planning and management", JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT, vol. 136, no. 4, 12 October 2009 (2009-10-12), pages 412-432, XP55021686, DOI: 10.1061/(ASCE)WR.1943-5452.0000053 the whole document	1-15
X,P	D. Hadka, P. Reed: "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization", Evolutionary Computation (Early Access), 4 October 2011 (2011-10-04), XP55021733, DOI: 10.1162/EVCO_a_00053 Retrieved from the Internet: URL:http://www.coe.psu.edu/water/images/f/f0/Hadka_&_Reed_In-Press_MOEA_Diagnostics.pdf [retrieved on 2011-10-05] section 2	1-15
X,P	D. Hadka: "MOEA Framework version 1.8", Sourceforge, 21 November 2011 (2011-11-21), XP55021673, Retrieved from the Internet: URL:http://sourceforge.net/projects/moeaframework/files/MOEAFramework-1.8/ [retrieved on 2012-12-03] files "Instrumenter.java", "NEWS" & "faqs.html"	1-15
X,P	J. Kollat, P. Reed: "Evolutionary multiobjective optimization in water resources: The past, present, and future", Presentation at Aerospace Corporation, 10 January 2012 (2012-01-10), XP55021736, Retrieved from the Internet: URL:http://www.coe.psu.edu/water/images/archive/2/24/20120227182426!Kollat_AeroCorp2012_MOEAComparison.pdf [retrieved on 2012-02-27] slides 35-43	1-15

-/--

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/022972

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
T	D. Hadka, P. Reed: "Borg: An auto-adaptive many-objective evolutionary computing framework", Evolutionary Computation (Just Accepted), 2 March 2012 (2012-03-02), XP55021663, DOI: 10.1162/EVCO_a_00075 Retrieved from the Internet: URL:http://www.coe.psu.edu/water/images/1/1d/Hadka&Reed.InPress.Borg.pdf [retrieved on 2012-03-02] sections 2 & 3 -----	
L	P. M. Reed: "Curriculum vitae", Hydroinformatics Research Group (Pennsylvania State University, College of Engineering), 11 October 2011 (2011-10-11), XP55021664, Retrieved from the Internet: URL:http://www.coe.psu.edu/water/images/1/1a/Reed.CV.pdf [retrieved on 2011-10-11] paragraph "Patents" -----	