

Программные системы для оценки качества Парето-аппроксимации в задаче многокритериальной оптимизации. Обзор

04, апрель 2014

DOI: 10.7463/0414.0709198

Белоус В. В., Грошев С. В., профессор, д.ф.-м.н. Карпенко А. П., Шибитов И. А.

УДК 519.6

Россия, МГТУ им. Баумана

valentina.belous@gmail.com

groshev_sergev@mail.ru

karpenko@rk6.bmstu.ru

vanoshibitov91@mail.ru

Введение

Классические методы решения задачи многокритериальной оптимизации (МКО) основаны на сведении ее к совокупности задач глобальной однокритериальной оптимизации. Относительно новый и быстро развивающийся класс МКО-методов образуют методы Парето-аппроксимации, предполагающие предварительное построение некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето МКО-задачи.

Известно большое число методов Парето-аппроксимации, что порождает проблему оценки сравнительной эффективности этих методов. Такую оценку естественно производить на основе оценки качества полученной Парето-аппроксимации и суммарных вычислительных затрат. Таким образом, задача оценки эффективности методов Парето-аппроксимации сама является, как минимум, двухкритериальной.

В содержательных терминах качество Парето-аппроксимации может быть оценено с помощью следующих характеристик: близость найденных решений к точному множеству Парето рассматриваемой МКО-задачи; равномерность распределения решений в полученной Парето-аппроксимации; мощность найденного множества решений. Качество Парето-аппроксимации можно оценивать также путем сравнения найденных решений с некоторыми другими, например, «эталонными» решениями.

В практически значимых МКО-задачах, в которых, как правило, используется более двух критериальных функций, визуальный анализ качества соответствующей Парето-аппроксимации затруднен. Поэтому разработано большое число критериев (индикаторов) качества Парето-аппроксимации, формализующих указанные характеристики.

В работе даем постановку МКО-задачи, приводим краткий обзор методов решения этой задачи на основе ее Парето-аппроксимации, представляем основные известные инди-

каторы качества этих методов. Основное содержание работы представляет собой обзор известных программных систем, ориентированных на решение задачи Парето-аппроксимации, которые реализуют различные алгоритмы оценки качества этой аппроксимации

1. Постановка задачи многокритериальной оптимизации

Множеством допустимых значений вектора варьируемых параметров X является ограниченное и замкнутое множество

$$D_X = \{X \mid G(X) \geq 0\} \subset \{X\} = R^{|X|}.$$

Критериальная вектор-функция $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ со значениями в пространстве критериев $\{F\} = R^{|F|}$ определена в области D_X . Лицо, принимающее решения (ЛПР), стремится минимизировать в этой области каждый из частных критериев оптимальности $f_1(X), f_2(X), \dots, f_{|F|}(X)$, что условно записываем в виде

$$\min_{X \in D_X} F(X) = F(X^*) = F^*, \quad (1)$$

где векторы X^* , F^* - искомое решение МКО-задачи.

Критериальная вектор-функция $F(X)$ выполняет отображение множества D_X в некоторое множество D_F пространства критериев, которое называется множеством достижимости. Введем на множествах D_F , D_X отношение доминирования. Говорим, что вектор $F_1 = F(X_1) \in D_F$ доминирует вектор $F_2 = F(X_2) \in D_F$, что записываем в виде $F_1 \succ F_2$, если среди равенств и неравенств $f_k(X_1) \leq f_k(X_2)$, $k \in [1:|F|]$ имеется, хотя бы одно строгое. Вектор X_1 доминирует вектор X_2 ($X_1 \triangleright X_2$), если вектор $F(X_1)$ доминирует вектор $F(X_2)$.

Выделим из множества D_F подмножество точек D_F^* - фронт Парето МКО-задачи (1), которые не доминируются другими точками этого множества и среди которых нет доминирующих друг друга. Множество $D_X^* \in D_X$, соответствующее множеству D_F^* , называется множеством Парето указанной МКО-задачи. Таким образом, если $X \in D_X^*$, то $F(X) \in D_F^*$.

2. Методы решения МКО-задачи на основе Парето-аппроксимации

Методы решения МКО-задачи чрезвычайно разнообразны. Существует несколько способов классификации этих методов. Используем классификацию, в соответствие с которой выделяют методы зондирования, априорные, апостериорные, адаптивные алгоритмы, а также методы Парето-аппроксимации [1].

Выделяем непопуляционные и популяционные методы Парето-аппроксимации. В отличие от непопуляционных методов, в популяционных методах реализуется одновременный поиск не одной, а большого числа недоминируемых точек, близких к множеству или к фронту Парето.

Обычно популяционные методы Парето-аппроксимации строят на основе эволюционных и, чаще всего, генетических алгоритмов. Принципиальными в этом случае являются правила формирования фитнес-функции, обеспечивающей перемещение индивидов популяции, в конечном счете, в направлении множества Парето. Эволюция же этих индивидов может протекать по законам не только эволюционных, но и других популяционных алгоритмов [2].

Обозначим Θ^F , Θ^X архивные множества, содержащие не доминируемые точки F_j^Θ и соответствующие им точки X_j^Θ ; $j \in [1:|\Theta|]$. Большинство популяционных методов Парето-аппроксимации использует итерационное уточнение множеств точек в архивах Θ^F , Θ^X . Если при этом на итерации t появляется новая точка F_i , доминирующая некоторые точки из архива Θ^F , то все доминируемые точки, а также соответствующие им точки из архива Θ^X , удаляем. При удовлетворении некоторого критерия останова, текущее содержимое архивов Θ^F , Θ^X полагаем искомой аппроксимацией фронта D_F^* и множества Парето D_X^* соответственно.

В популяционных методах Парето-аппроксимации новые точки для архивов Θ^F , Θ^X «поставляет» популяция S агентов s_i , текущие координаты которых в пространстве поиска равны X_i , а в целевом пространстве - $F_i = F(X_i)$; $i \in [1:|S|]$.

Основной проблемой построения популяционных методов Парето-аппроксимации является конструирование *фитнес-функции*, обеспечивающей перемещение агентов популяции s_i в направлении множества Парето D_X^* , а соответствующих точек F_i - в направлении фронта Парето D_F^* . В силу, как правило, меньшей размерности критериального пространства $\{F\}$ по сравнению с размерностью пространства параметров $\{X\}$, ответ на вопрос о направлении и шаге перемещения агентов обычно отыскивают в терминах пространства $\{F\}$, а не пространства $\{X\}$. Важно также, что относительно фронта Парето, в отличие от множества Парето, имеется некоторая априорная информация [2].

По способу определения приспособленности агентов выделяем популяционные методы Парето-аппроксимации на основе *лексикографической селекции*, *чередующихся целевых функций* и на основе *ранжирования* агентов. Последние методы непосредственно используют концепцию доминирования по Парето и нашли наибольшее применение в вычислительной практике [2].

3. Индикаторы качества Парето аппроксимации

Такие характеристики качества Парето-аппроксимации, как близость найденных решений к точному множеству Парето рассматриваемой МКО-задачи, равномерность распределения решений в полученной Парето-аппроксимации, мощность найденного множества решений, оценивают с помощью *унарных* индикаторов качества (*unary quality indicators*). Оценку качества Парето-аппроксимации на основе сравнения найденных недоминируемых решений с некоторыми «эталонными» решениями выполняют с помощью бинарных индикаторов (*binary quality indicators*) [3].

Далее для простоты записи в обозначениях архивного множества Θ^F решений, недоминируемых в пространстве $\{F\}$, опускаем индекс F , и для элементов множества Θ используем обозначения вида $F_j, j \in [1:|\Theta|]$.

Унарные индикаторы. Среднее расстояние до точного фронта (*Generalization Distance, GD*) определяет формула

$$I_{GD}(\Theta) = \frac{\left(\sum_{j=1}^{|\Theta|} \|F_j, F_j^*\|_E^{1/|F|} \right)^{1/|F|}}{|\Theta|} \rightarrow \min,$$

где F_j^* - ближайшая к точке F_j точка множества D_F^* . Индикатор характеризует близость найденных решений к точному фронту Парето D_F^* .

Среднее рассеяние (*Spacing, S*) $I_S(\Theta)$ представляет собой меру равномерности распределения решений, полученных в результате Парето-аппроксимации. Индикатор задает формула

$$I_S(\Theta) = \sqrt{\frac{1}{|\Theta|-1} \sum_{j=1}^{|\Theta|} \text{abs}(\bar{d} - d_j)} \rightarrow \min, \quad (2)$$

где $d_j = \min_{k \in [1:|\Theta|], k \neq j} \|F_j, F_k\|_M$ - минимальное манхеттоновское расстояние [3]

между архивным решением F_j и остальными архивными решениями; $\bar{d}$ - среднее всех этих величин.

Максимальное рассеяние (*Maximum Spread, MS*) $I_{MS}(\Theta)$ определяет формула, аналогичная формуле (2):

$$I_{MS}(\Theta) = \sqrt{\sum_{j=1}^{|\Theta|} \max_{k \in [1:|\Theta|]} \|F_j, F_k\|_M} \rightarrow \min.$$

Отклонение от равномерного распределения (*Deviation from Uniform distribution, DU*) $I_{DU}(\Theta)$, аналогично индикаторам $I_S(\Theta)$, $I_{MS}(\Theta)$, определяет равномерность распределения архивных недоминируемых решений Θ в целевом пространстве. Индикатор задает формула

$$I_{DU}(\Theta) = \frac{1}{|\Theta|} \sum_{j=1}^{|\Theta|} \text{abs}(d_j - \bar{d}),$$

где d_j - евклидово расстояние решения F_j до ближайшего из решений множества Θ ; $\bar{d}$ - среднее этих величин.

Мощность множества решений (*Overall Nondominated Vector Generation*) $I_{ONVG}(\Theta)$ есть ни что иное, как число элементов множества Θ , то есть

$$I_{ONVG}(\Theta) = |\Theta| \rightarrow \max.$$

Объем объемлющего гиперкуба (*Hypercube enclosing indicator, HC*) $I_{HC}(\Theta)$ определяют величины

$$I_{HC_1}(\Theta) = \max_{P \in R^{|F|}} \left(P \succ F_j, \forall j \in [1:|\Theta|] \right) = p, \quad P = (p, p, \dots, p),$$

$$I_{HC_2}(\Theta) = \min_{Q \in R^{|F|}} \left(Q \succ F_j, \forall j \in [1:|\Theta|] \right) = q, \quad Q = (q, q, \dots, q).$$

Относительное число ошибочных решений (*Error Ratio, ER*) $I_{ER}(\Theta)$ - относительное число решений архивного множества Θ , не принадлежащих точному фронту Парето D_F^* рассматриваемой МКО-задачи:

$$I_{ER}(\Theta) = \frac{1}{|\Theta|} \sum_{j=1}^{|\Theta|} e_j \rightarrow \min, \quad e_j = \begin{cases} 0, & F_j \notin D_F^*, \\ 1, & F_j \in D_F^*. \end{cases}$$

Максимальная ошибка аппроксимации (*Maximum Pareto front Error, MPE*) $I_{MPE}(\Theta)$ также для своего вычисления требует знания точного фронта Парето. Индикатор определяет формула

$$I_{MPE}(\Theta) = \max_{j \in [1:|\Theta|]} \|F_j, F_j^*\|_M^{|F|} \rightarrow \min,$$

где, как и ранее, F_j^* - ближайшая к точке F_j точка множества D_F^* .

Значительное число унарных индикаторов качества Парето-аппроксимации можно построить на основе понятий ε -доминирования и ε_+ -доминирования [3].

Бинарные индикаторы. Определим бинарный индикатор $I_\varepsilon(\Theta_1, \Theta_2)$ ($I_{\varepsilon_+}(\Theta_1, \Theta_2)$), формализующий близость множеств недоминируемых решений Θ_1, Θ_2 , как минимальную величину ε , при которой для любого вектора $F_2 \in \Theta_2$ существует вектор $F_1 \in \Theta_1$, который ε (ε_+)-доминирует вектор F_2 , то есть такой вектор, что $F_1 \succ_\varepsilon F_2$ ($F_1 \succ_{\varepsilon_+} F_2$).

Положим, что множество $\Theta_*^F = \Theta_*$ недоминируемых решений в пространстве D_F является «эталоном», а множество $\Theta^F = \Theta$ - аналогичным оцениваемым множеством.

ε -индиктор $I_\varepsilon(\Theta_*, \Theta)$ строят на основе индикатора $I_\varepsilon(\Theta_1, \Theta_2)$. Величина $I_\varepsilon(\Theta_*, \Theta)$ формализует близость множества Θ к множеству Θ_* как минимальную величину ε , при которой для любых векторов $F_j \in \Theta$, $F_k \in \Theta_*$ имеет место отношение $F_j \succ_\varepsilon F_k$.

$\mathcal{E}_+$ -индикатор $I_{\mathcal{E}_+}(\Theta_*, \Theta)$ определяем аналогичным образом.

Индикатор $I_C(\Theta_*, \Theta)$, называемый покрытием (*Coverage*), может быть использован в том случае, когда множество Θ_* является счетным. Величина $I_C(\Theta_*, \Theta)$ имеет смысл относительного числа решений, принадлежащих множеству Θ_* , которые доминируются некоторыми из решений множества Θ :

$$I_C(\Theta_*, \Theta) = \frac{\left| \left\{ F_k \in \Theta_* \mid \exists F_j \in \Theta : F_j \succ F_k \right\} \right|}{|\Theta_*|} \rightarrow \max.$$

Здесь числитель представляет собой число указанных решений.

Следующий бинарный индикатор строят на основе величины, которая называется разностью покрытий (*coverage differences*). Пусть Θ_1, Θ_2 - множества недоминируемых векторов данной МКО-задачи. Величина

$$\delta(\Theta_1, \Theta_2) = \zeta(\Theta_1 + \Theta_2) - \zeta(\Theta_2)$$

называется разностью покрытий и представляет собой некоторую меру $\zeta(\cdot)$ части множества достижимости, слабо доминируемой множеством Θ_1 , но не слабо доминируемой множеством Θ_2 [3].

Индикатор гиперобъема разности покрытий (*HyperVolume of coverage difference*) $I_{HV}(\Theta_*, \Theta)$ в качестве меры $\zeta(\cdot)$ использует объем соответствующей части множества достижимости.

Ряд критериев качества Парето-аппроксимации построен Хансеном (*M. P. Hansen*) и Жажкевичем (*A. Jaszkiewicz*) на основе функции предпочтений ЛПР (*decision-maker's utility function*) $u(F)$, которая осуществляет отображение множества достижимости D_F во множество вещественных чисел, то есть $u: R^{|F|} \rightarrow R^1$, и формализует предпочтения ЛПР на множестве D_F , так что большим значениям этой функции соответствуют лучшие с его точки зрения решения [4].

4. Обзор программных систем МКО-оптимизации

Разделяем указанные программные системы на те, которые поддерживают оценку качества Парето-аппроксимации, и те, которые эти оценки не поддерживают.

4.1. Системы, реализующие оценку качества Парето-аппроксимации

Система Opt4j представляет собой модульный фреймворк метаэвристической оптимизации [5]. Система разработана подразделением *Hardware/Software Co-Desing* университета *Erlang en-Nuremberg* (Германия), является свободно распространяемой и написана на языке программирования *Java*. Система ориентирована на решение сложных задач оптимизации посредством их декомпозиции на подзадачи, решение которых может произ-

водиться параллельно. Для поддержки модульности разработки используется шаблон проектирования *Dependency Injection (DI)*.

Задачу оптимизации полностью определяет реализация интерфейсов *Creator*, *Decoder*, *Evaluator* (рисунок 1). Алгоритм оптимизации реализуется с помощью интерфейса оптимизатора (*Optimizer*), который манипулирует популяцией (*Population*) и архивом (*Archive*). Популяция содержит особей (*Individuals*) текущей итерации, а архив — недоминируемых особей, полученных в течение всего процесса оптимизации.

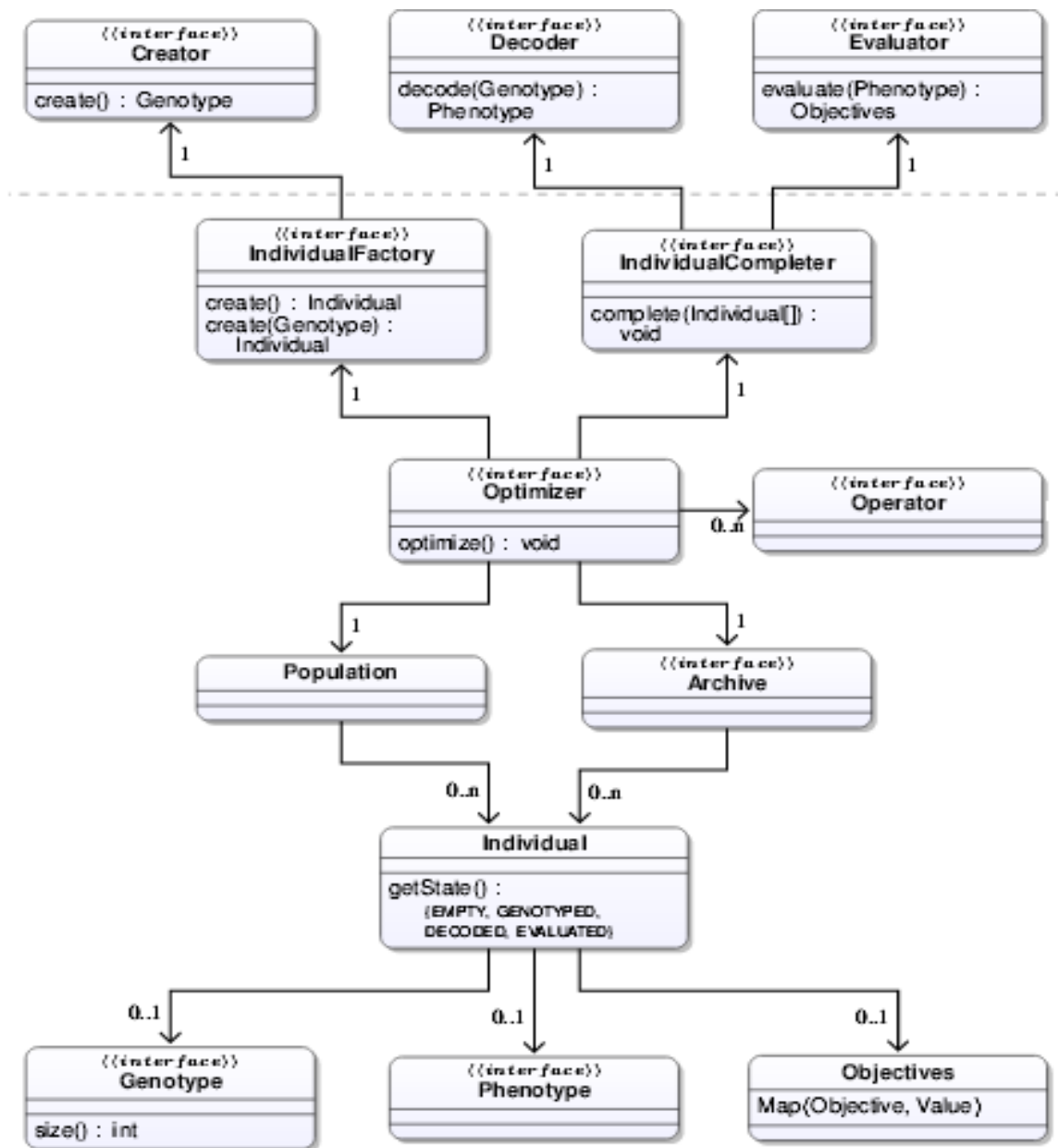


Рисунок 1. Структура классов системы *Opt4j*: блоки, расположенные выше пунктирной линии, относятся к задаче оптимизации; ниже ее – к методу оптимизации.

Система *Opt4j* использует следующие сторонние библиотеки.

- *Google Guice (Dependency Injection Framework)* - универсальная библиотека

с открытым исходным кодом для Java-платформы, разработанная компанией Google под лицензией Apache 2.0. Библиотека обеспечивает поддержку внедрения зависимостей при помощи аннотаций для конфигурирования объектов Java [6].

- SAT4J ([SATisfiability Library for Java](#)) - решатель для задач выполнимости [7].
- [Ptolemy Plot \(Plot Library for Java\)](#) - средство создания 2D графики [8].

В настоящее время *Opt4j* содержит набор реализаций таких методов МКО-оптимизации как *SPEA2* [9], *NSGA2* [10], *SMS* [11], *MOPSO* [12]. В систему включены известные тестовые МКО-задачи *ZDT*, *DTLZ* [13], *WFG* [14], *LOTZ* [15]. Реализованы критерии оценки качества решений *Hypervolume*, *Spread*, *Epsilon*, *Generational Distance*, *Inverted Generational Distance* [2, 16].

К достоинствам системы *Opt4j* можно отнести фокусированность на МКО-задачи, использование технологии аспектно-ориентированного программирования *Google Guice* для управления зависимостями и объединения компонентов воедино. Недостатком системы является отсутствие WEB-интерфейса и полноценной системы оценки качества Парето-аппроксимации.

Система jMetal также является свободно распространяемым фреймворком. Система ориентирована на разработку и изучение метаэвристических МКО-методов [17]. Помимо этих методов система включает в себя реализации однокритериальных генетических алгоритмов, методы эволюционных стратегий и роя частиц, а также алгоритма дифференциальной эволюции.

В настоящее время в системе представлены следующие методы:

- метаэвристические методы многокритериальной оптимизации *NSGA-II*, *ssNSGA-II*, *GDE3* [18], *PAES* [19], *SMS-EMOA* [20], *MOEA/D* [21], *SPEA2*, *PESA-II*, *OMOPSO* [11], *MOCHC* [22], *IBEA* [23], *FastPGA* [24], *DENSEA* [25], *CellDE* [26];
- метаэвристические методы однокритериальной оптимизации *generational GA* (*gGA*), *steady-state GA* (*ssGA*) [27], *DE*, *PSO* и *CMA-ES* [28];
- тестовые задачи однокритериальной и многокритериальной оптимизации *Fonseca*, *Kursawe*, *Schaffer*, *Srinivas*, *Tanaka*, *Rastrigin*, *Rosenbrock*, *ZDT*, *DTLZ*, *LZ09* [16];
- индикаторы оценки качества Парето-аппроксимации *Hypervolume*, *Spread*, *Epsilon*, *Generational Distance*, *Inverted Generational Distance*;
- алгоритмы сбора и анализа статистики;
- алгоритмы визуализации результатов.

В системе *jMetal* реализованы также оригинальные авторские МКО-методы *SMPSO* (*Speed constrained Multi-objective PSO*) [29], *AbYSS* (*Archive-based hYbrid Scatter Search*) [30], *MOCeLL* (*MultiObjective Cellular genetic algorithm*) [31]. Система портирована на языки программирования Java (*jMetal*), C++ (*jMetalC++*) и C# (*jMetal.Net*).

К достоинствам системы *jMetal* можно отнести ориентацию на многокритериальную оптимизацию, реализацию 15 современных многокритериальных эволюционных алгоритмов. Недостатками системы являются отсутствие средств автоматизированного запуска задач и WEB-интерфейса.

Система PISA [32] состоит из двух следующих компонентов.

1) Текст-ориентированный интерфейс для алгоритмов поисковой оптимизации, предполагающий разделение задачи оптимизации на два модуля. Первый модуль (*Variator*) содержит все части, специфичные для конкретной задачи оптимизации. Второй модуль (*Selector*) включает в себя независимые от задачи оптимизации части. Модули обмениваются данными посредством текстового файла и модуля *Monitor* (рисунок 2).

2) Библиотека готовых к использованию модулей: тестовых задач, методов оптимизации и модулей для оценки производительности (*PISA Performance Assessment*, *PISA PA*).

Схему взаимодействия указанных компонентов иллюстрирует рисунок 3.



Рисунок 2. Схема текст-ориентированного интерфейса системы *jMetal*

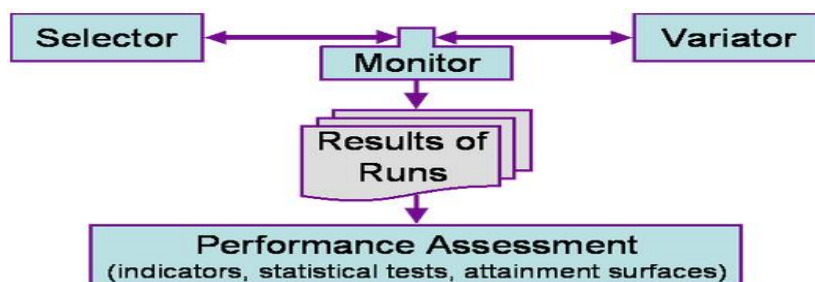


Рисунок 3. Схема взаимодействия компонентов системы *jMetal*

Модуль *Monitor* отвечает за многократные запуски различных методов на разных тестовых задачах, сохранение промежуточных и конечных результатов работы этих методов. В частности, модуль контролирует коммуникации между модулями *Variator*, *Selector* с целью сохранения информации о генерируемых популяциях, осуществляет автоматический вызов различных методов оптимизации для получения входных данных для последующего статистического анализа, отслеживает состояния процессов *Monitor* и *Selector*.

Основным назначением модулей *PISA PA* является оценка эффективности и сравнение МКО-методов на различных тестовых и реальных задачах [32]. Для оценки эффективности методов Парето-аппроксимации могут быть использованы индикаторы *Hypervolume*, *Spread*, *Epsilon*.

Система *PISA* не имеет *WEB* интерфейса.

MOEA Framework [33] представляет собой свободно распространяемый набор библиотек, написанных на языке программирования *Java*. Набор распространяется под лицензией *GNU Lesser General Public License*. Помимо МКО-методов в *MOEA* реализованы средства для быстрой разработки, запуска и статистического анализа результатов. Ключевые особенности данного проекта состоят в следующем.

- Набор реализованных методов практически идентичен набору, реализованному в системе *jMetal*. Возможен импорт программ из систем *jMetal* и *PISA*.
- Имеется возможность расширять наборы методов оптимизации, тестовых задач и операторов собственными реализациями.
- Модульная архитектура позволяет средствами *Service Provider Interface (SPI)* комбинировать существующие компоненты, создавая новые методы оптимизации.
- Система включает в себя более 1100 тестовых задач, для проверки адекватности решений [34].
- Реализованы критерии оценки качества решений *Hypervolume*, *Generational Distance*, *Inverted Generational Distance*, *Additive ε -indicator*, *Spacing*.

- Поддерживается статистический анализ результатов.

В системе *MOEA*, в отличие от других известных систем, реализована поддержка анализа чувствительности (*sensitivity analysis*) – исследование того, как значения свободных параметров метода влияют на индикаторы его эффективности [35]. На основе этого анализа может быть выполнена метаоптимизация данного метода [36]. Система не имеет *WEB*-интерфейса.

4.2. Системы, не поддерживающие оценку качества Парето-аппроксимации

Система *ParadisEO* – это многофункциональный объектно-ориентированный программный комплекс, написанный на языке программирования *C++* [37, 38]. В данный момент система портирована на операционные системы *Linux*, *MacOS*, *Windows*. Текущая версия системы включает в себя реализацию следующих компонентов:

- эволюционные МКО-методы *MOGA*, *NSGA*, *NSGA-II*, *SPEA2*, *IBEA*, *SEEA*;
- МКО-методы локального поиска *PLS* (*Pareto Local Search*), *IBMOLS* [39], *DMLS* [40];
- гибридные, параллельные и распределенные версии указанных методов;
- тестовые задачи *ZDT*, *DTLZ*;
- комбинаторные задачи коммивояжера (*TSP*), назначения (*Quadratic Assignment Problem*, *QAP*), планирования (*flowshop*);
- реальные задачи выбора маршрута (*routing*), планирования (*scheduling*), биоинформатики.

В настоящее время в системе реализована только модель распараллеливания мастер\подчиненный с поддержкой прогрессивного [41] и линейного [42] методов балансировки.

В системе не реализованы критерии оценки качества Парето аппроксимаций, отсутствует *WEB*-интерфейс.

Система *Watchmaker Framework* – расширяемый высокопроизводительный объектно-ориентированный фреймворк для реализации платформо-независимых эволюционных и генетических алгоритмов на языке программирования *Java* [43-45]. Система является свободно распространяемым продуктом под лицензией *Apache Software Licence*.

Основными являются следующие отличительные особенности продукта.

- Многопоточное эволюционное ядро для повышения производительности вычислений на многопроцессорных и многоядерных компьютерах.
- Встраиваемые стратегии селекции. Помимо уже реализованных стратегий возможна реализация стратегий, определенных пользователем.
- Интерактивные эволюционные алгоритмы.
- Распределенная обработка данных с поддержкой распределенных хранилищ данных *Hadoop* (в рамках проектах *Apache Mahout*) и *Terracotta*.
- Средства для визуализации и создания интерфейса пользователя (*Evolution Monitor*).

К преимуществам системы можно отнести достаточно широкий набор нестандартных тестовых задач *Mona Lisa* [43], *Sudoku* [44], *Biomorphs* [45] и т.д. Из недостатков следует отметить ориентированность на однокритериальные задачи и отсутствие *WEB*-интерфейса.

ЕСJ представляет собой ориентированную на исследования *Java* библиотеку, разработанную в *Evolutionary Computation Laboratory* университета *George Mason* (Виргиния, США) [46]. В системе реализован ряд эволюционных методов, включая МКО-методы

NSGA-II, *SPEA-II*. Реализована поддержка островной и мастер\подчиненный моделей параллелизма. Реализация островной модели поддерживает асинхронную работу с сокетами поверх стека протоколов *TCP/IP*. Среди недостатков системы следует отметить ориентацию на однокритериальные методы оптимизации, отсутствие реализаций оценки качества решений, отсутствие *WEB*-интерфейса.

Другие системы. Для полноты картины назовем также системы *JGAP* [47], *JCLEC* [48], *JavaEvA* [49], *GAA* [50], *Jenes* [51], *EO* [52], *Open Beagle* [53], *Heuristic Lab* [54, 55], имеющие в настоящее время недостаточно развитую функциональность.

Заключение

На основании представленного обзора программных систем можно сделать следующие выводы.

В силу высокой актуальности задач многокритериальной оптимизации ведется интенсивная разработка программных систем, поддерживающих все аспекты решения таких задач, включая оценку качества Парето-аппроксимации. Указанные программные системы являются модульными, часто, свободно распространяемыми. Широко используются готовые библиотеки апробированных прикладных программ.

Программные системы поддерживают, как правило, большое число известных и новых методов Парето-аппроксимации, а также алгоритмов оценки качества этой аппроксимации. Широко представлены в системах тестовые задачи многокритериальной оптимизации. Типичным является реализация широкого набора алгоритмов сбора и анализа статистики, алгоритмов визуализации результатов.

Поскольку практические задачи многокритериальной оптимизации имеют высокую вычислительную сложность, программные системы включают в себя параллельные методы Парето-аппроксимации, ориентированные на различные классы параллельных вычислительных систем. Имеется возможность использовать различные алгоритмы балансировки загрузки этих систем.

Некоторые программные системы поддерживают анализ чувствительности - анализ того, как значения свободных параметров метода Парето-аппроксимации влияют на индикаторы его эффективности. На основе этого анализа может быть выполнена метаоптимизация методов Парето-аппроксимации.

Ни одна из представленных программных систем не имеет *WEB*-интерфейса, что можно считать их общим недостатком.

Список литературы

1. Лотов А.В., Поспелова И.И. Многокритериальные задачи принятия решений: учеб. пособие. М.: МАКС Пресс, 2008. 197 с.
2. Карпенко А.П., Семенихин А.С., Митина Е.В. Популяционные методы аппроксимации множества Парето в задаче многокритериальной оптимизации // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 4. Режим доступа: <http://www.technomag.edu.ru/doc/363023.html> (дата обращения 01.03.2014).
3. Белоус В.В., Грабик А.В., Грошев С.В., Шибитов И.А. Качество Парето-аппроксимации в задаче многокритериальной оптимизации // XVIII Байкальская Все-

- русская конференция «Информационные и математические технологии в науке и управлении»: материалы. Ч. 1. Иркутск: ИСЭМ СО РАН, 2013. С. 6-12.
4. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation*. 2000. Vol. 8, no. 2. P. 173-195.
 5. Lukasiewicz M., Glass M., Reimann F., Teic J. Opt4J - A Modular Framework for Metaheuristic Optimization // *Proceedings of the Genetic and Evolutionary Computing Conference*, 2011. P. 1723-1730.
 6. Google Guice. Available at: <https://code.google.com/p/google-guice/> , accessed 01.03.2014.
 7. SAT4j. Available at: <http://www.sat4j.org/> , accessed 01.03.2014.
 8. Ptplot. Available at: <http://ptolemy.eecs.berkeley.edu/java/ptplot/> , accessed 01.03.2014.
 9. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization // In: *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)* / K. Giannakoglou, et al., editors. International Center for Numerical Methods in Engineering (CIMNE), 2002. P. 95-100.
 10. Agrawal S., Pratap A., Meyarivan T., Deb K. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II // In: *Parallel Problem Solving from Nature PPSN VI*. Springer Berlin Heidelberg, 2000. P. 849-858. DOI: [10.1007/3-540-45356-3_83](https://doi.org/10.1007/3-540-45356-3_83)
 11. Reyes Sierra M., Coello Coello C.A. Improving PSO-based Multi-Objective Optimization using Crowding, Mutation and e-Dominance // In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005. P. 505-519. DOI: [10.1007/978-3-540-31880-4_35](https://doi.org/10.1007/978-3-540-31880-4_35)
 12. Emmerich M., Beume N., Naujoks B. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion // In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005. P. 62-76. DOI: [10.1007/978-3-540-31880-4_5](https://doi.org/10.1007/978-3-540-31880-4_5)
 13. Deb K., Thiele L., Laumanns M., Zitzler E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Tech. Rep. 112. Zurich, Switzerland, 2001. 27 p.
 14. Barone L., While L., Huband L., Hingston S. Use of the WFG Toolkit and PISA for Comparison of MOEAs // *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 2007. P. 382-389. DOI: [10.1109/MCDM.2007.369117](https://doi.org/10.1109/MCDM.2007.369117)
 15. Laumanns M., Thiele L., Zitzler E. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions // *IEEE Transactions on Evolutionary Computation*. 2004. Vol. 8, no. 2. P. 170-182. DOI: [10.1109/TEVC.2004.823470](https://doi.org/10.1109/TEVC.2004.823470)

16. Zitzler E., Thiele L., Laumanns M., Fonseca C.V., Fonseca V.G. Performance Assessment of Multiobjective Optimizers: An Analysis and Review // IEEE Transactions of Evolutionary Computation. 2003. Vol. 7, no. 2. P. 117-132. DOI: [10.1109/TEVC.2003.810758](https://doi.org/10.1109/TEVC.2003.810758)
17. Durillo J.J., Nebro A.J. jMetal: A Java Framework for Multi-Objective Optimization // Advances in Engineering Software. 2011. Vol. 42. P. 760-771.
18. Kukkonen S., Lampinen J. GDE3: The Third Evolution Step of Generalized Differential Evolution. KanGAL Report No. 2005013, 2005. P. 443-450.
19. Knowles J., Corne D. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization // Proceedings of the 1999 Congress on Evolutionary Computation, Piscataway, NJ, 1999. P. 98-105.
20. Beume N. SMS-EMOA: Multiobjective selection based on dominated hypervolume // European Journal of Operational Research. 2007. Vol. 181, no. 3. P. 1653-1669.
21. Li H., Zhang Q. Multiobjective Optimization problems with Complicated Pareto Sets, MOEA/D and NSGA-II // IEEE Transactions on Evolutionary Computation. 2009. Vol. 13, no. 2. P. 284-302. DOI: [10.1109/TEVC.2008.925798](https://doi.org/10.1109/TEVC.2008.925798)
22. Nebro O. Optimal Antenna Placement using a New Multiobjective CHC Algorithm // Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, 2007. P. 876-883.
23. Zitzler E., Kunzli S. Indicator-based selection in multi objective search // In: Parallel Problem Solving from Nature - PPSN VIII. Springer Berlin Heidelberg, 2004. P. 832-842. DOI: [10.1007/978-3-540-30217-9_84](https://doi.org/10.1007/978-3-540-30217-9_84) (Ser. Lecture Notes in Computer Science; vol. 3242).
24. Eskandari H., Geiger C.D., Lamont G.B. FastPGA: A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems // In: Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2007. P. 141-155. DOI: [10.1007/978-3-540-70928-2_14](https://doi.org/10.1007/978-3-540-70928-2_14)
25. Greiner D., Emperador J.M., Winter G. Enhancing the multiobjective optimum design of structural trusses with evolutionary algorithms using DENSEA // In: 44th AIAA (American Institute of Aeronautics and Astronautics) Aerospace Sciences Meeting and Exhibit, 2006. Paper AIAA-2006-1474 (11 p).
26. Durillo J.J., Nebro A.J., Luna F., Alba E. Solving Three-Objective Optimization Problems. Using a new Hybrid Cellular Genetic Algorithm // In: Parallel Problem Solving from Nature - PPSN X. Springer Berlin Heidelberg, 2008. P. 661-670. DOI: [10.1007/978-3-540-87700-4_66](https://doi.org/10.1007/978-3-540-87700-4_66)
27. Vavak F., Fogarty T.C. Comparison of Steady State and Generational Genetic Algorithms for Use in Nonstationary Environments // Proc. of the IEEE International Conference on Evolutionary Computation, 1996. P. 192-195. DOI: [10.1109/ICEC.1996.542359](https://doi.org/10.1109/ICEC.1996.542359)
28. Hansen N.R., Ros N., Mauny M., Auger S.A. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems // Applied Soft Computing. 2011. Vol. 11. P. 5755-5769.

29. Nebro J., Durillo J.J., Coello Coello C.A. Analysis of Leader Selection Strategies in a Multi-Objective Particle Swarm Optimizer // 2013 IEEE Congress on Evolutionary Computation, 2013. P. 3153 - 3160. DOI: [10.1109/CEC.2013.6557955](https://doi.org/10.1109/CEC.2013.6557955)
30. Nebro A.J., Luna F., Alba E., Dorronsoro B., Durillo J.J., Beham A. AbYSS. Adapting Scatter Search to Multiobjective Optimization // IEEE Transactions on Evolutionary Computation. 2006. Vol. 12, no. 4. P. 439-457. DOI: [10.1109/TEVC.2007.913109](https://doi.org/10.1109/TEVC.2007.913109)
31. Nebro J., Durillo J.J., Luna F., Dorronsoro B., Alba E. MOCell. A Cellular Genetic Algorithm for Multiobjective Optimization // International Journal of Intelligent Systems. 2009. Vol. 24, no. 7. P. 726-746.
32. Bleuler S., Laumanns M., Thiele L., Zitzler E. PISA—A Platform and Programming Language Independent Interface for Search Algorithms // In: Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2003. P. 494–508. DOI: [10.1007/3-540-36970-8_35](https://doi.org/10.1007/3-540-36970-8_35)
33. MOEA Framework. Available at: <http://moeaframework.org/documentation.html> , accessed 01.03.2014.
34. Knowles J.D., Thiele L., Zitzler E. A tutorial on the performance assessment of the stochastic multiobjective optimizers. TIK-Report No. 214. Computer Engineering and Network Laboratory, ETH Zurich, 2006. 35 p.
35. Hadka D., Reed P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization // Evolutionary Computation. 2012. Vol. 20, no. 3. P. 423-452.
36. Karpenko A.P., Sviaadze Z.O. Meta-optimization based on self-organizing map and genetic algorithm // Optical Memory and Neural Networks (Information Optics). 2011. Vol. 20, no.4. P. 279-283.
37. ParadisEO. Available at: <http://paradiseo.gforge.inria.fr> , accessed 01.03.2014.
38. Liefvooghe L., Jourdan T., Legrand J., Talbi G. ParadisEO-MOEO: A Software Framework for Evolutionary Multi-objective Optimization // In: Advances in Multi-objective Nature Inspired Computing. Springer Berlin Heidelberg, 2010. P. 87-117. DOI: [10.1007/978-3-642-11218-8_5](https://doi.org/10.1007/978-3-642-11218-8_5) (Ser. Studies in Computational Intelligence; vol. 272).
39. Basseur M., Burke E.K. Indicator-based multi-objective local search // IEEE Congress on Evolutionary Computation (CEC'2007), 2007. P. 3100-3107. DOI: [10.1109/CEC.2007.4424867](https://doi.org/10.1109/CEC.2007.4424867)
40. Liefvooghe J., Humeau S., Mesmoudi S., Jourdan L. On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems // Journal of Heuristics. 2011. Vol. 18, iss. 2. P. 317-352. DOI: [10.1007/s10732-011-9181-3](https://doi.org/10.1007/s10732-011-9181-3)
41. Huang J., Zhao F., Chen J., Pei J., Yin J. Towards Progressive and Load Balancing Distributed Computation: A Case Study on Skyline Analysis // Journal of Computer Science and Technology. 2010. Vol. 25, no. 3. P. 431-443.

42. Christophe D., Raphael C., Fabienne J. Load balancing of the direct linear multisplitting method in a grid computing environment. Tech. rep. LIFC, 2008. 29 p.
43. Watchmaker Framework for Evolutionary Computation. Evolving the Mona Liza. Available at: <http://watchmaker.uncommons.org/examples/monalisa.php> , accessed 01.03.2014.
44. Watchmaker Framework for Evolutionary Computation. An Evolutionary Sudoku Solver. Available at: <http://watchmaker.uncommons.org/examples/sudoku.php> , accessed 01.03.2014.
45. Watchmaker Framework for Evolutionary Computation. Watchmaker. Biomorphs. Available at: <http://watchmaker.uncommons.org/examples/biomorphs.php> , accessed 01.03.2014.
46. A Java-based Evolutionary Computation Research System. Available at: <http://cs.gmu.edu/~eclab/projects/ecj/> , accessed 01.03.2014.
47. Java Genetic Algorithm Package. Available at: <http://jgap.sourceforge.net/> , accessed 01.03.2014.
48. JCLEC - Java Class Library for Evolutionary Computation. Available at: <http://jclec.sourceforge.net/> , accessed 01.03.2014.
49. A Java-based framework for Evolutionary Algorithms (Java/EvA). Available at: <http://www.ra.cs.uni-tuebingen.de/software/JavaEvA/> , accessed 01.03.2014.
50. Genetic Algorithm Playground. Available at: www.aridolan.com/ga/gaa/gaa.html , accessed 01.03.2014.
51. Jenes. Genetic Algorithms in Java. Available at: <http://jenes.intelligentia.it/> , accessed 01.03.2014.
52. Evolving Objects (EO): An Evolutionary Computation Framework. Available at: <http://eodev.sourceforge.net/> , accessed 01.03.2014.
53. Parizeau G.M. Genericity in Evolutionary Computation Software Tools: Principles and Case Study // International Journal on Artificial Intelligence Tools. 2006. Vol. 15, no. 2. P. 173-194.
54. HeuristicLab: Paradigm-independent and Extensible Environment for Heuristic Optimization. Available at: <http://dev.heuristiclab.com/trac/hl/core> , accessed 01.03.2014.
55. Wagner S., Kronberger G., Beham A., Kommenda M., Scheibenpflug A., Pitzer E., Vonolfen S., Kofler M., Winkler S., Dorfer V., Affenzeller M. Architecture and Design of the HeuristicLab Optimization Environment // In: Advanced Methods and Applications in Computational Intelligence. Springer International Publishing, 2014. P. 197-261. DOI: [10.1007/978-3-319-01436-4_10](https://doi.org/10.1007/978-3-319-01436-4_10) (Ser. Topics in Intelligent Engineering and Informatics; vol. 6).

Programme systems to estimate the Pareto-approximation quality in the problem of multi-criteria optimization. A review.

04, April 2014

DOI: 10.7463/0414.0709198

V.V. Belous, S. V. Groshev, A.P. Karpenko, I. A. Shibitov

Bauman Moscow State Technical University, 105005, Moscow, Russian Federation

valentina.belous@gmail.com

groshev_sergey@mail.ru

karpenko@rk6.bmstu.ru

vanoshibitov91@mail.ru

The paper considers a relatively new and rapidly growing class of multi-criteria optimization (MCO) methods. This class is based on the preliminary creation of a finite-dimensional approximation of the Pareto set and, subsequently, the Pareto-approximation of MCO problem. Since many different methods are known, there is a problem to compare these methods with each other. Solving real world MCO problems one often faces more than two criterion functions. This fact makes it difficult to perform a visual analysis of the proper Pareto-approximation. Therefore,

A lot of quality indicators of Pareto-approximation have been developed. In this work a statement of multi-criteria optimization problem was presented along with a brief review of the specified class of methods and existent quality indicators. The paper itself is devoted to the review of existent programme systems, designed for building Pareto-approximation, which implement various algorithms of quality estimation.

Publications with keywords: [multiobjective optimization](#), [quality estimation of Pareto-approximation](#), [programme system](#)

Publications with words: [multiobjective optimization](#), [quality estimation of Pareto-approximation](#), [programme system](#)

References

1. Lotov A.V., Pospelova I.I. *Mnogokriterial'nye zadachi prinyatiya resheniy* [Multicriteria decision making problems]. Moscow, MAKS Press, 2008. 197 p. (in Russian).

2. Karpenko A.P., Semenikhin A.S., Mitina E.V. [Review: population methods of Pareto set approximation in multi-objective optimization problem]. *Nauka i obrazovanie MGTU im. N.E. Bauman - Science and Education of the Bauman MSTU*, 2012, no. 4. Available at: <http://technomag.edu.ru/en/doc/363023.html> , accessed 01.03.2014. (in Russian).
3. Belous V.V., Grabik A.V., Groshev S.V., Shibitov I.A. [Quality of Pareto-approximation in multicriteria optimization problem]. *18 Baykal'skaya Vserossiyskaya konferentsiya "Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii": materialy* [Proc. of the 18th Baikal all-Russian conference "Information and Mathematical Technologies in Science and Management"]. Pt. 1. Irkutsk, Publ. of Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences (ESI SB RAS), 2013, pp. 6-12. (in Russian).
4. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 2000, vol. 8, no. 2, pp. 173-195.
5. Lukasiwycz M., Glass M., Reimann F., Teic J. Opt4J - A Modular Framework for Metaheuristic Optimization. *Proceedings of the Genetic and Evolutionary Computing Conference*, 2011. P. 1723-1730.
6. Google Guice. Available at: <https://code.google.com/p/google-guice/> , accessed 01.03.2014.
7. SAT4j. Available at: <http://www.sat4j.org/> , accessed 01.03.2014.
8. Ptplot. Available at: <http://ptolemy.eecs.berkeley.edu/java/ptplot/> , accessed 01.03.2014.
9. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou K., et al., eds. *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*. International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95-100.
10. Agrawal S., Pratap A., Meyarivan T., Deb K. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: *Parallel Problem Solving from Nature PPSN VI*. Springer Berlin Heidelberg, 2000, pp. 849-858. DOI: [10.1007/3-540-45356-3_83](https://doi.org/10.1007/3-540-45356-3_83)
11. Reyes Sierra M., Coello Coello C.A. Improving PSO-based Multi-Objective Optimization using Crowding, Mutation and e-Dominance. In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005, pp. 505-519. DOI: [10.1007/978-3-540-31880-4_35](https://doi.org/10.1007/978-3-540-31880-4_35)
12. Emmerich M., Beume N., Naujoks B. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005, pp. 62-76. DOI: [10.1007/978-3-540-31880-4_5](https://doi.org/10.1007/978-3-540-31880-4_5)

13. Deb K., Thiele L., Laumanns M., Zitzler E. *Scalable Test Problems for Evolutionary Multi-Objective Optimization*. Tech. Rep. 112. Zurich, Switzerland, 2001. 27 p.
14. Barone L., While L., Huband L., Hingston S. Use of the WFG Toolkit and PISA for Comparison of MOEAs. *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 2007, pp. 382-389. DOI: [10.1109/MCDM.2007.369117](https://doi.org/10.1109/MCDM.2007.369117)
15. Laumanns M., Thiele L., Zitzler E. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions. *IEEE Transactions on Evolutionary Computation*, 2004, vol. 8, no. 2, pp. 170–182. DOI: [10.1109/TEVC.2004.823470](https://doi.org/10.1109/TEVC.2004.823470)
16. Zitzler E., Thiele L., Laumanns M., Fonseca C.V., Fonseca V.G. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions of Evolutionary Computation*, 2003, vol. 7, no. 2, pp. 117-132. DOI: [10.1109/TEVC.2003.810758](https://doi.org/10.1109/TEVC.2003.810758)
17. Durillo J.J., Nebro A.J. jMetal: A Java Framework for Multi-Objective Optimization. *Advances in Engineering Software*, 2011, vol. 42, pp. 760-771.
18. Kukkonen S., Lampinen J. *GDE3: The Third Evolution Step of Generalized Differential Evolution*. KanGAL Report No. 2005013, 2005, pp. 443-450.
19. Knowles J., Corne D. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ, 1999, pp. 98-105.
20. Beume N. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007, vol. 181, no. 3, pp. 1653-1669.
21. Li H., Zhang Q. Multiobjective Optimization problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2009, vol. 13, no. 2, pp. 284-302. DOI: [10.1109/TEVC.2008.925798](https://doi.org/10.1109/TEVC.2008.925798)
22. Nebro O. Optimal Antenna Placement using a New Multiobjective CHC Algorithm. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, London, England, 2007, pp. 876-883.
23. Zitzler E., Kunzli S. Indicator-based selection in multi objective search. In: *Parallel Problem Solving from Nature - PPSN VIII*. Springer Berlin Heidelberg, 2004, pp. 832-842. DOI: [10.1007/978-3-540-30217-9_84](https://doi.org/10.1007/978-3-540-30217-9_84) (Ser. *Lecture Notes in Computer Science*; vol. 3242).
24. Eskandari H., Geiger C.D., Lamont G.B. FastPGA: A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems. In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2007, pp. 141-155. DOI: [10.1007/978-3-540-70928-2_14](https://doi.org/10.1007/978-3-540-70928-2_14)

25. Greiner D., Emperador J.M., Winter G. Enhancing the multiobjective optimum design of structural trusses with evolutionary algorithms using DENSEA. In: *44th AIAA (American Institute of Aeronautics and Astronautics) Aerospace Sciences Meeting and Exhibit*, 2006, paper AIAA-2006-1474 (11 p).
26. Durillo J.J., Nebro A.J., Luna F., Alba E. Solving Three-Objective Optimization Problems. Using a new Hybrid Cellular Genetic Algorithm. In: *Parallel Problem Solving from Nature - PPSN X*. Springer Berlin Heidelberg, 2008, pp. 661-670. DOI: [10.1007/978-3-540-87700-4_66](https://doi.org/10.1007/978-3-540-87700-4_66)
27. Vavak F., Fogarty T.C. Comparison of Steady State and Generational Genetic Algorithms for Use in Nonstationary Environments. *Proc. of the IEEE International Conference on Evolutionary Computation*, 1996, pp. 192-195. DOI: [10.1109/ICEC.1996.542359](https://doi.org/10.1109/ICEC.1996.542359)
28. Hansen N.R., Ros N., Mauny M., Auger S.A. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems. *Applied Soft Computing*, 2011, vol. 11, pp. 5755-5769.
29. Nebro J., Durillo J.J., Coello Coello C.A. Analysis of Leader Selection Strategies in a Multi-Objective Particle Swarm Optimizer. *2013 IEEE Congress on Evolutionary Computation*, 2013, pp. 3153 - 3160. DOI: [10.1109/CEC.2013.6557955](https://doi.org/10.1109/CEC.2013.6557955)
30. Nebro A.J., Luna F., Alba E., Dorronsoro B., Durillo J.J., Beham A. AbYSS. Adapting Scatter Search to Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 2006, vol. 12, no. 4, pp. 439-457. DOI: [10.1109/TEVC.2007.913109](https://doi.org/10.1109/TEVC.2007.913109)
31. Nebro J., Durillo J.J., Luna F., Dorronsoro B., Alba E. MOCeLl. A Cellular Genetic Algorithm for Multiobjective Optimization. *International Journal of Intelligent Systems*, 2009, vol. 24, no. 7, pp. 726-746.
32. Bleuler S., Laumanns M., Thiele L., Zitzler E. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In: *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2003, pp. 494–508. DOI: [10.1007/3-540-36970-8_35](https://doi.org/10.1007/3-540-36970-8_35)
33. MOEA Framework. Available at: <http://moeaframework.org/documentation.html> , accessed 01.03.2014.
34. Knowles J.D., Thiele L., Zitzler E. *A tutorial on the performance assessment of the stochastic multiobjective optimizers*. TIK-Report No. 214. Computer Engineering and Network Laboratory, ETH Zurich, 2006. 35 p.
35. Hadka D., Reed P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evolutionary Computation*, 2012, vol. 20, no. 3, pp. 423-452.

36. Karpenko A.P., Svianadze Z.O. Meta-optimization based on self-organizing map and genetic algorithm. *Optical Memory and Neural Networks (Information Optics)*, 2011, vol. 20, no.4, pp. 279-283.
37. ParadisEO. Available at: <http://paradiseo.gforge.inria.fr> , accessed 01.03.2014.
38. Liefvooghe L., Jourdan T., Legrand J., Talbi G. ParadisEO-MOEO: A Software Framework for Evolutionary Multi-objective Optimization. In: *Advances in Multi-objective Nature Inspired Computing*. Springer Berlin Heidelberg, 2010. P. 87-117. DOI: [10.1007/978-3-642-11218-8_5](https://doi.org/10.1007/978-3-642-11218-8_5) (Ser. *Studies in Computational Intelligence*; vol. 272).
39. Basseur M., Burke E.K. Indicator-based multi-objective local search. *IEEE Congress on Evolutionary Computation (CEC'2007)*, 2007, pp. 3100-3107. DOI: [10.1109/CEC.2007.4424867](https://doi.org/10.1109/CEC.2007.4424867)
40. Liefvooghe J., Humeau S., Mesmoudi S., Jourdan L. On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *Journal of Heuristics*, 2011, vol. 18, iss. 2, pp. 317-352. DOI: [10.1007/s10732-011-9181-3](https://doi.org/10.1007/s10732-011-9181-3)
41. Huang J., Zhao F., Chen J., Pei J., Yin J. Towards Progressive and Load Balancing Distributed Computation: A Case Study on Skyline Analysis. *Journal of Computer Science and Technology*, 2010, vol. 25, no. 3, pp. 431-443.
42. Christophe D., Raphael C., Fabienne J. *Load balancing of the direct linear multisplitting method in a grid computing environment*. Tech. rep. LIFC, 2008. 29 p.
43. Watchmaker Framework for Evolutionary Computation. Evolving the Mona Liza. Available at: <http://watchmaker.uncommons.org/examples/monalisa.php> , accessed 01.03.2014.
44. Watchmaker Framework for Evolutionary Computation. An Evolutionary Sudoku Solver. Available at: <http://watchmaker.uncommons.org/examples/sudoku.php> , accessed 01.03.2014.
45. Watchmaker Framework for Evolutionary Computation. Watchmaker. Biomorphs. Available at: <http://watchmaker.uncommons.org/examples/biomorphs.php> , accessed 01.03.2014.
46. A Java-based Evolutionary Computation Research System. Available at: <http://cs.gmu.edu/~eclab/projects/ecj/> , accessed 01.03.2014.
47. Java Genetic Algorithm Package. Available at: <http://jgap.sourceforge.net/> , accessed 01.03.2014.
48. JCLEC - Java Class Library for Evolutionary Computation. Available at: <http://jclec.sourceforge.net/> , accessed 01.03.2014.
49. A Java-based framework for Evolutionary Algorithms (Java/EvA). Available at: <http://www.ra.cs.uni-tuebingen.de/software/JavaEvA/> , accessed 01.03.2014.

50. Genetic Algorithm Playground. Available at: www.aridolan.com/ga/gaa/gaa.html , accessed 01.03.2014.
51. Jenes. Genetic Algorithms in Java. Available at: <http://jenes.intelligentia.it/> , accessed 01.03.2014.
52. Evolving Objects (EO): An Evolutionary Computation Framework. Available at: <http://eodev.sourceforge.net/> , accessed 01.03.2014.
53. Parizeau G.M. Genericity in Evolutionary Computation Software Tools: Principles and Case Study. *International Journal on Artificial Intelligence Tools*, 2006, vol. 15, no. 2, pp. 173-194.
54. HeuristicLab: Paradigm-independent and Extensible Environment for Heuristic Optimization. Available at: <http://dev.heuristiclab.com/trac/hl/core> , accessed 01.03.2014.
55. Wagner S., Kronberger G., Beham A., Kommenda M., Scheibenpflug A., Pitzer E., Vonolfen S., Kofler M., Winkler S., Dorfer V., Affenzeller M. Architecture and Design of the HeuristicLab Optimization Environment. In: *Advanced Methods and Applications in Computational Intelligence*. Springer International Publishing, 2014. P. 197-261. DOI: [10.1007/978-3-319-01436-4_10](https://doi.org/10.1007/978-3-319-01436-4_10) (Ser. *Topics in Intelligent Engineering and Informatics*; vol. 6).