

УДК 004.41

EDN: [PYSNZK](#)



Модифицированный алгоритм роя частиц в задачах компоновки модулей отказоустойчивых программных систем

И.В. Ковалев^{1,2,3,4}, Д.И. Ковалев^{1,2*}, Т.П. Мансурова¹, Е.А. Борисова¹

¹Красноярский краевой Дом науки и техники РосСНИО, Красноярск, Россия

²Красноярский государственный аграрный университет, Красноярск, Россия

³Сибирский федеральный университет, Красноярск, Россия

⁴Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, Красноярск, Россия

*E-mail: grimm7jow@gmail.com

Аннотация. В статье представлена модификация алгоритма роя частиц для решения задачи компоновки модулей отказоустойчивых программных систем. Решение данной задачи актуально при синтезе структурно-сложных систем мониторинга траектории полета воздушных судов в экстремальных условиях Арктики и Крайнего Севера. Представлен стандартный алгоритм и вариант его модификации, в рамках которого вводится переменная, обозначающая инерционную массу, то есть влияние скорости в предыдущий момент времени контролируется показателем инерции. Чем больше показатель инерции, тем больше способность алгоритма к глобальному поиску, чем меньше, тем больше способность алгоритма к локальному поиску. Представленная модификация алгоритма роя частиц позволяет решать задачу компоновки модулей отказоустойчивых программных систем путем подбора подмножеств программных компонент из исходного множества, обеспечивая максимизацию целевой функции. Решена основная проблема при переносе PSO алгоритма на задачу данного типа, заключающаяся в обеспечении связного и непрерывного движения частиц в дискретном пространстве допустимых решений.

Ключевые слова: алгоритм роя частиц, отказоустойчивость, программная система, мониторинг, модуль, критерий поиска

Modified particle swarm algorithm in problems of assembly of modules for fault-tolerant software systems

I.V. Kovalev^{1,2,3,4}, D.I. Kovalev^{1,2*}, T.P. Mansurova¹, E.A. Borisova¹

¹Krasnoyarsk Regional Science and Technology City Hall, Krasnoyarsk, Russia

²Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia

³Siberian Federal University, Krasnoyarsk, Russia

⁴Siberian State University of Science and Technology named after academician M.F. Reshetnev, Krasnoyarsk, Russia

Annotation. The article presents a modification of the particle swarm algorithm for solving the problem of assembling modules of fault-tolerant software systems. The solution of this problem is relevant in the synthesis of structurally complex systems for monitoring the flight path of aircraft in the extreme conditions of the Arctic and the Far North. A standard algorithm and a variant of its modification are presented, within the framework of which a variable denoting the inertial mass is introduced, that is, the influence of the speed at the previous moment of time is controlled by the inertia indicator. The larger the inertia index, the greater the ability of the algorithm to global search, the smaller, the greater the ability of the algorithm to local search. The presented modification of the particle swarm algorithm allows solving the problem of assembling modules of fault-tolerant software systems by selecting subsets of software components from the original set, ensuring the maximization of the objective function. The main problem is solved when transferring the PSA algorithm to a problem of this type, which consists in providing a connected and continuous motion of particles in a discrete space of feasible solutions.

Keywords: particle swarm algorithm, fault tolerance, software system, monitoring, module, search criterion

1. Введение

В работах [1-4] показана применимость алгоритма роя частиц (particle swarm optimization, PSO) для решения задач компоновки модулей отказоустойчивых программных систем. Отметим, что решение данной задачи актуально при синтезе структурно-сложных систем мониторинга траектории полета воздушных судов в экстремальных условиях Арктики и Крайнего Севера [5]. В рамках стандартного алгоритма подразумевается, что каждая частица определяется с помощью текущей скорости и позиции, а также локального лучшего решения частицы и глобального лучшего решения, что отражают следующие уравнения:

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1^k (pBest_{id}^k - x_{id}^k) + c_2 r_2^k (gBest_{id}^k - x_{id}^k); \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (2)$$

Здесь v_{id}^k и x_{id}^k - скорость частицы i в момент времени k и координаты ее позиции соответственно; $pBest_{id}^k$ - координаты в области допустимых решений, относящиеся к лучшему решению, найденному частицей i к моменту времени k ; $gBest_{id}^k$ - координаты лучшего решения, найденного всеми частицами в d -мерном пространстве к моменту времени k . Скорость частицы в каждом из пространств ограничивается между $-v_{dMax}$ и v_{dMax} . Переменные c_1 и c_2 являются регуляторами скорости, отвечающими за приоритет при выборе направления движения. Значения c_1 и c_2 , как правило, приравнивают к 2. Переменные r_1^k и r_2^k представляют из себя случайные отклонения, обычно расположенные в интервале $[0, 1]$.

2. Модифицированный алгоритм роя частиц

Рассмотрим уравнение (1). Его правая часть состоит из трех частей: первая часть представляет собой информацию о скорости частицы в предыдущий момент времени, вторая и третья часть отвечают за изменение скорости частицы. Без них частица будет двигаться в одном направлении с постоянной скоростью до тех пор, пока не достигнет границы допустимой области. Алгоритм роя частиц не сможет найти приемлемое решение, если только оно не присутствует на траектории ее движения. Если мы рассмотрим уравнение (1) без первой части, то мы получим, что скорость движения частиц зависит только от их текущей позиции и лучших значений, которые были найдены в процессе работы алгоритма.

Предположим, что в начале работы алгоритма, частица i находится в глобальной лучшей позиции, тогда скорость движения частицы будет равна 0, так как значения локальной и глобальной лучшей позиции равны. Так будет продолжаться до тех пор, пока другая частица не найдет решение, которое превосходит лучшее глобальное решение. В это же время другие частицы будут двигаться в точку равновесия между глобальным лучшим решением и локальным лучшим решением, найденным данной частицей. Как упоминалось ранее рекомендуемые значения для c_1 и c_2 равно 2, что делает влияние персональных и общих знаний частиц на процесс принятия решения о направлении движения одинаковым. Поэтому частицы в процессе алгоритма стягиваются в область лучшего глобального решения до тех пор, пока не будет найдено другое лучшее решения, и процесс повторяется.

Из этого можно сделать вывод, что область, в которой происходит поиск решения, уменьшается в процессе работы алгоритма. Этот процесс виден более явно, при его компьютерной визуализации [2]. Можно будет увидеть, что при отсутствии первой части в уравнении (1), все частицы будут двигаться в направлении одной точки, и область поиска будет уменьшаться с каждой итерацией. Только в случае, когда глобальное лучшее решение было в начальной области поиска, появляется вероятность того, что алгоритм роя частиц сможет его найти. Конечный результат будет целиком зависеть от начального расположения частиц в области поиска [3].

Однако с добавлением первой части в уравнение, у частиц появляется возможность расширять первоначальную область поиска, что позволяет им находить новые, возможно более качественные, решения. Таким образом, после добавления первой части уравнения у частиц появляется способность к глобальному поиску. Учет этого позволяет внести следующие изменения в стандартный алгоритм роя частиц, представленные в уравнениях (3) и (4).

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1^k (pBest_{id}^k - x_{id}^k) + c_2 r_2^k (gBest_{id}^k - x_{id}^k) \quad (3)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (4)$$

Вводится переменная ω , обозначающая инерционную массу. Влияние скорости в предыдущий момент времени контролируется показателем инерции. Чем больше показатель инерции, тем больше способность алгоритма к глобальному поиску, чем меньше, тем лучше способность алгоритма к локальному поиску. В общем случае, при $\omega = 1$, в более позднем периоде работы алгоритма присутствует недостаточная

способность алгоритма к локальному поиску. Экспериментально было установлено, что алгоритм роя частиц демонстрирует большую скорость сходимости, когда $\omega \in [0.8, 1.2]$. Также было установлено, что линейное уменьшение показателя ω от 0,9 до 0,4 позволяет быстро обнаружить область решения в начале поиска, а в дальнейшем обнаружить оптимальное решение более точно. В процессе уменьшения показателя инерционности замедляется скорость движения частицы, чтобы сделать сам процесс поиска более точным в области нахождения решения. Данная модификация позволяет увеличить скорость сходимости, и решение, получаемое таким способом, обладает большей точностью. При решении сложных задач компоновки мультиверсионных модулей программных систем это улучшение позволяет повысить качество принимаемого решения на поздних этапах поиска.

3. Применение PSA алгоритма для задач компоновки подмножества модулей

В стандартном алгоритме PSO каждая частица знает о своих координатах в области поиска, также частица знает о координатах лучшего решения, найденного до сих пор. Алгоритм хранит значение функции, связанное с лучшим найденным решением. Другим значением, которое хранит каждая частица, является координата и значение лучшего решения, которое было найдено каждой частицей независимо друг от друга. На каждой итерации работы PSO алгоритма изменяется скорость каждой частицы, с ориентацией на позиции локального и глобального лучшего решения, известного частице. Также задействуется фактор случайности. Это достигается благодаря случайным переменным, обеспечивающим смену приоритета между лучшим локальным и лучшим глобальным решением. В общем схема работы PSO алгоритма представлена ниже.

Инициализация положения и скорости каждой частицы

do

for каждой частицы

оценить значение целевой функции в текущей точке

end

for каждой частицы

обновить значение персонального лучшего решения

end

обновить значение глобального лучшего решения

for каждой частицы
 обновить скорость
 рассчитать новые координаты
end

while (проверка на соответствие критерию останова)

На начальном этапе развития PSO алгоритмов основным типом задач, для которых он применялся, были задачи нахождения экстремумов на непрерывных функциях [6]. В настоящее время увеличивается количество работ по исследованию возможности применения алгоритма роя частиц для решения более сложных задач [7-9].

Задача составления (компоновки) подмножества модулей отказоустойчивых программных систем ставит своей целью подбор подмножеств программных компонент из исходного множества, дающих максимальное значение целевой функции. В отличие от задачи коммивояжера (TSP - traveling salesman problem) возможные решения представляют из себя подмножества разных параметров.

Для того чтобы решить задачу составления подмножества с помощью алгоритма роя частиц, каждый объект в рое содержит в себе D -размерный вектор, где $D = |M|$ и равно размеру данного множества M . Каждое измерение в векторе представлено бинарным битом и определяет, используется ли соответствующий элемент для подмножества, которое будет принято, как решение задачи. Основной проблемой при переносе алгоритма на задачу такого типа является обеспечение связного и непрерывного движения частиц в дискретном пространстве допустимых решений.

В так называемом дискретном бинарном алгоритме роя частиц (discrete binary PSO algorithm) это делается следующим образом. Так как область допустимых решений является дискретной, и частица не должна находиться в одной и той же позиции, используется компонент случайности при вычислении новой позиции. Основной идеей является предположение считать для высокой скорости наличие высокой вероятностью того, что связанный с ней бит информации принимает значение 1.

Формально значение скорости частицы определяется также как и в уравнении (3). Для того чтобы определить вероятность использования компонента в итоговом решении, используется функция, которая связывает значение скорости со значением в интервале $[0, 1]$. Используется следующая функция:

$$sig(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \quad (5)$$

Для того чтобы определить значение i -го бита в векторе частицы d , генерируется случайное число r_{id} в интервале $[0, 1]$, а затем, в случае если $r_{id} < sig(v_{id})$, значение устанавливается равным 1, в противном случае, значение устанавливается равным 0.

4. Заключение

Таким образом, представленная модификация алгоритма роя частиц позволяет решать задачу компоновки модулей отказоустойчивых программных систем путем подбора подмножеств программных компонент из исходного множества, обеспечивая максимизацию целевой функции. Решена основная проблема при переносе PSO алгоритма на задачу данного типа, заключающаяся в обеспечении связного и непрерывного движения частиц в дискретном пространстве допустимых решений.

Благодарности

Проведение исследований осуществляется при поддержке КГАУ «Красноярский краевой фонд поддержки научной и научно-технической деятельности» в рамках проекта «Контроль траектории полета воздушных судов в экстремальных условиях Арктики и Крайнего Севера» в соответствии с заявкой 2021110907918.

Список литературы

1. Соловьев, Е. В. Использование метода роя частиц для формирования состава мультиверсионного программного обеспечения / И. В. Ковалев, Д. И. Ковалев, К. К. Бахмарева, А. В. Демиш // Приборы и системы. Управление, контроль, диагностика. – 2013. – № 3. – С. 1-6.
2. Соловьев, Е. В. Использование метода роя частиц для формирования состава мультиверсионного программного обеспечения / Е. В. Соловьев // Молодёжь и наука: Сборник материалов VIII Всероссийской научно-технической конференции студентов, аспирантов и молодых учёных, посвященной 155-летию со дня рождения К. Э. Циолковского [Электронный ресурс]. – Красноярск: Сибирский федеральный ун-т, 2012. <http://conf.sfu-kras.ru/sites/mn2012/section12.html>.
3. Карпенко, А. П. Ко-эволюционный алгоритм глобальной оптимизации на основе алгоритма роя частиц / А. П. Карпенко, Е. Ю. Воробьева // Машиностроение и компьютерные технологии. – 2013. – № 11. – С. 431-474.
4. Зенюткин, Н. В. О способах формирования информационных структур для моделирования объектов, сред и процессов / Н. В. Зенюткин, Д. И. Ковалев,

- Е. В. Туев, Е. В. Туева // Современные инновации, системы и технологии. Modern Innovations, Systems and Technologies. – 2021. – № 1(1). – С. 10-22. <https://doi.org/10.47813/2782-2818-2021-1-1-10-22>.
5. Серова, Н. А. Транспортная инфраструктура Российской Арктики: специфика функционирования и перспективы развития / Н. А. Серова, В. А. Серова // Проблемы прогнозирования. – 2021. – № 2 (185). – С. 142-151.
 6. Карпенко, А. П. Обзор методов роя частиц для задачи глобальной оптимизации (particle swarm optimization) / А. П. Карпенко, Е. Ю. Селиверстов // Машиностроение и компьютерные технологии. – 2009. – № 3. – С. 2-28.
 7. Акзигитов, Р. А. Повышение эффективности мониторинга воздушных судов посредством комплексной системы обнаружения объектов / Р. А. Акзигитов, А. В. Кацура, А. Р. Акзигитов, Д. Е. – Строкков: «Вестник СибГАУ». – 2016. – № 17(2). – С. 388-392.
 8. Карцан, И. Н. Построение наземных пунктов управления космическими аппаратами с использованием оптимизационно-имитационной модели / И. Н. Карцан // Современные инновации, системы и технологии. Modern Innovations, Systems and Technologies. – 2021. – № 1(2). – С. 64-71. <https://doi.org/10.47813/2782-2818-2021-1-2-64-71>.
 9. Ковалев, И. В. К вопросу формирования блочно-модульной структуры системы управления беспилотных летательных объектов / И. В. Ковалев, В. В. Лосев, М. В. Сарамуд, А. О. Калинин, А. С. Лифарь // Современные инновации, системы и технологии. Modern Innovations, Systems and Technologies. – 2021. – №1(3). – С. 48-64. <https://doi.org/10.47813/2782-2818-2021-1-3-48-64>.