

Раздел IX. Вычислительные комплексы нового поколения и нейрокомпьютеры

УДК 681.3

Ю.О. Чернышев, Н.Н. Венцов, П.А. Панасенко

ПРИМЕНЕНИЕ PSO-МЕТОДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ*

В работе сформулирована многомерная нелинейная задача распределения вычислительных ресурсов. На основе метода «оптимизации с использованием роя частиц» (Particle Swarm Optimization, PSO) разработан алгоритм решения поставленной задачи. Определена связь параметров задачи и решающего её алгоритма. Построено распределение результатов работы алгоритма по итерациям. На основании анализа распределения результатов работы алгоритма по итерациям сделано заключение о способности алгоритма выходить из областей локальных оптимумов. Приведены графики погрешностей алгоритма на задачах размерности 1–10 с шагом 1 и 5–100 с шагом 5. Экспериментально определена линейная вычислительная сложность алгоритма на размерностях 1–100.

Вычислительные ресурсы; коллективная адаптация; интеллектуальные системы; оптимизация.

Yu.O. Chernyshev, N.N. Vencov, P.A. Panasenko

APPLICATION OF PSO-METHODS FOR SOLVING ALLOCATION OF COMPUTATIONAL RESOURCES

In this paper we formulated a multi-dimensional nonlinear problem of distribution of computing resources. On the basis of "optimization using particle swarm» (Particle Swarm Optimization, PSO) algorithm to solve this problem. Defined relationship between the parameters of the problem and solve it the algorithm. We construct a distribution of the results of iterations of the algorithm. Based on the analysis of the distribution of the results of iterations of the algorithm was suggested algorithm is able to withdraw from areas of local optima. The graphs of the errors on the tasks dimension algorithm 1–10 in increments of 1, and 5–100 in increments of 5. The experimentally determined the computational complexity of the algorithm on the dimensions of 1–100.

Computing resources; collective adaptation; intelligent systems; optimization.

Постановка задачи. Имеется множество вычислительных ресурсов S , подлежащее распределению между N информационными процессами. Потери вычислительного ресурса при его диспетчеризации (выделении) j -му информационному процессу, определяются величиной p_j , $j \in \{1, 2, 3, \dots, N\}$. Необходимый минимум ресурсов требуемый j -му информационному процессу определяется величиной ξ_j , а максимум – ζ_j . Прибыльность j -го информационного процесса зависит от объема доступных вычислительных ресурсов x_j и описывается нелинейной функцией $f_j(x_j)$. Предположим, что ресурсы выделяются последовательно и однократно для каждого информационного процесса. Необходимо распределить ресурсы S таким образом, чтобы максимизировать получаемую прибыль, которая определяется по формуле:

* Работа выполнена при финансовой поддержке РФФИ (проект 12-01-00474).

$$\sum_{j=1}^N f_j(x_j) \rightarrow \max. \quad (1)$$

Ограничения, накладываемые на условия и решения задачи, описываются выражениями:

$$\sum_{j=1}^N (x_j + p_j) \leq S, \quad (2)$$

$$\sum_{j=1}^N \zeta_j > S, \quad (3)$$

$$\sum_{j=1}^N \xi_j < S, \quad (4)$$

$$\xi_j < x_j < \zeta_j, j \in \{1, 2, 3, \dots, N\}, \quad (5)$$

$$\xi_j > 0, j \in \{1, 2, 3, \dots, N\}. \quad (6)$$

Из неравенства (5) следует, что любому информационному процессу должно быть выделено некоторое количество ресурса ξ_j . Потери связанные с выделением данного ресурса составляют p_j . Таким образом, суммарные потери не зависят от распределения ресурсов и определяются как

$$P = \sum_{j=1}^N (\xi_j + p_j). \quad (7)$$

По этой причине объем фактически распределяемого ресурса \hat{S} определяется по формуле:

$$\hat{S} = S - P. \quad (8)$$

На основании выражений (1)–(5) можно считать, что решение задачи заключается в нахождении точки оптимума в N -мерном пространстве решений. На практике функционал (1) может не обладать свойствами, облегчающими его оптимизацию такими, как линейность, непрерывность, дифференцируемость в любой точке и т.д. По этой причине использование классических методов оптимизации не целесообразно.

Используемый подход. Для решения оптимизационных задач в настоящее время активно используются методы поиска оптимальных решений основанные на идеях самообучения, накопления и адаптации [1–4]. Реализация подобных подходов подразумевает наличие индивидуальной и коллективной составляющей процессов самообучения, накопления и адаптации [5–7]. Исследуем возможность применения метода «оптимизации с использованием роя частиц» (*Particle Swarm Optimization, PSO*) [7–9] для решения поставленной задачи. *PSO*-метод базируется на понятии популяции, и моделирует поведение птиц в стае и косяков рыб. Стратегия поведения особей (частиц) в популяции (рое) состоит в стремлении превзойти достижения соседних частиц и улучшить собственные. Выбор траектории движения осуществляются частицей на основе личного опыта, а также опыта её соседей.

Как утверждалось ранее, решение поставленной задачи состоит в нахождении точки оптимума в N -мерном пространстве решений. Если предположить что частица движется в N -мерном пространстве, то ее координаты можно описать вектором (кортежем) $X = \langle x_1, x_2, \dots, x_j, \dots, x_N \rangle$. Значение каждого элемента кортежа определяет количество ресурсов выделенных соответствующему информационному процессу, например величина x_1 определяет количество ресурсов выделенных первому процессу, x_2 – второму и т.д. С помощью кортежа $X_i(t) = \langle x_{i,1}(t), x_{i,2}(t), \dots, x_{i,j}(t), \dots, x_{i,N}(t) \rangle$ обозначим позицию i -ой частицы $i \in \{1, 2, 3, \dots, M\}$ в пространстве поиска решений в момент времени t . Для простоты изложения, факт вычисления функционала (1) на основе кортежа $X_i(t) = \langle x_{i,1}(t), x_{i,2}(t), \dots, x_{i,j}(t), \dots, x_{i,N}(t) \rangle$ обозначим как:

$$F[X_i(t)] \rightarrow \max. \quad (9)$$

Процесс поиска решений данным методом начинается с генерации частиц. Начальное состояние частицы i , в нулевой момент времени, описывается кортежем $X_i(0) = \langle x_{i,1}(0), x_{i,2}(0), \dots, x_{i,j}(0), \dots, x_{i,N}(0) \rangle$ и определяется следующим образом $X_i(0) = U(\xi, \zeta)$, где $\xi = \{ \xi_1, \xi_2, \dots, \xi_N \}$ и $\zeta = \{ \zeta_1, \zeta_2, \dots, \zeta_N \}$. Функция $U(\xi, \zeta)$ для каждого измерения генерирует случайное число из диапазона $[\xi_j, \zeta_j]$. Позиция i -ой частицы в пространстве поиска решений изменяется добавлением скорости $V_i(t) = \langle v_{i,1}(t), v_{i,2}(t), \dots, v_{i,N}(t) \rangle$ к текущей позиции:

$$X_i(t+1) = X_i(t) + V_i(t+1). \quad (10)$$

В разновидности *gbest* PSO-метода каждая частица тяготеет к лучшему решению целого роя поэтому скорость i -ой частицы в j -ом измерении определяется по формуле [7–9]:

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1,j}(t) [y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t) [y^*_{j}(t) - x_{i,j}(t)], \quad (11)$$

где $v_{i,j}(t)$ – скорость i -ой частицы в j -ом измерении в момент времени t ; $x_{ij}(t)$ – координаты частицы i в измерении j ; c_1 и c_2 – положительные константы ускорения, варьирующие когнитивную и социальную компоненты скорости частицы; r_{1j} и r_{2j} – случайные переменные, принимающие значения 0 или 1; $y_{i,j}(t)$ – координата наилучшей достигнутой позиции частицы i в j -ом измерении. $y^*_{j}(t)$ – координата наилучшей достигнутой позиции роя в j -ом измерении.

В соответствии с формулами (10) и (11) в случае если: $x_{i,j}(t+1) < \xi_j$ то $x_{i,j}(t+1) := \xi_j$, если же $x_{i,j}(t+1) > \zeta_j$ то $x_{i,j}(t+1) := \zeta_j$.

Кортеж $Y_i = \langle x_{i,1}, x_{i,2}, \dots, x_{i,N} \rangle$ отображает наилучшую позицию частицы i , которую она посещала, начиная с первой итерации. Обозначим функцию $\sum_{j=1}^N f_j(x_j)$ из выражения (1) через $F(X)$, тогда в момент времени $t+1$ следующая оптимальная позиция частицы i рассчитывается по формуле:

$$\begin{cases} Y_i = Y_i \text{ если } F(X(t+1)) \leq F(Y) \\ Y_i = X(t+1) \text{ если } F(X(t+1)) > F(Y) \end{cases} \quad (12)$$

Величина $\max(Y_i) \ i \in \{1, 2, 3, \dots, M\}$, где M число особей в рое, определяет наилучшее решение полученное роем.

Предлагаемый алгоритм. На основе изложенного выше метода был разработан эвристический алгоритм оптимизации распределения вычислительных ресурсов. Работа алгоритма начинается с выполнения блока команд под названием «Генерация особей», который описывает процесс создания особей, образующих пчелиный рой. Координаты генерируемых особей соответствуют ограничениям (2)–(6). Каждой особи ставится в соответствие значение целевой функции (1). Скорости всех сгенерированных особей равны нулю. В дальнейшем работа алгоритма состоит в итерационном выполнении следующих блоков: «Расчет статистики», «Расчет скоростей», «Перемещение особей», «Расчет ЦФ». Блок «Расчет статистики» описывает процесс нахождения величины $Y^*(t)$ – координаты наилучшей достигнутой позиции роя, найденной за t итераций, и величины $Y_i(t)$ – координаты наилучшей достигнутой позиции частицы, найденной за t итераций. Под наилучшей позицией понимается такое положение частицы в пространстве, при котором значение функционала (1) максимально. Таким образом, величина $Y^*(t)$ определяется для всего роя, а величина $Y_i(t)$ – для каждой частицы в отдельности. Блок «Расчет скоростей» описывает процесс расчета для каждой особи роя по формуле (11) новой скорости в каждом из пространств. Блок «Перемещение особей» описывает процесс расчета для каждой особи роя по формуле (10) новых координат. Блок «Расчет ЦФ» соответствует процессу вычисления целевой функции (1) для каждой особи роя, на основе новых координат. Критерием остановки алгоритма является выполнение заданного числа итераций.

Экспериментальная часть. Рассмотрим процесс поиска решения методом пчелиного роя в динамике (рис. 1).

На рис. 1 представлены две диаграммы «Глобальный оптимум» и «Текущий оптимум» оси абсцисс которых соответствуют итерациям разработанного алгоритма. Значения оси ординат диаграммы «Глобальный оптимум» соответствуют стоимости наилучшего решения найденного алгоритмом за все предшествующие итерации. По оси ординат диаграммы «Текущий оптимум» откладывается стоимость наилучшего решения найденного на текущей итерации. На 4 и 5 итерациях рис. 1 диаграммы «Текущий оптимум» алгоритм покидает зону локального оптимума, но затем, на 6–9 итерациях определяет более предпочтительное решение. Данный факт позволяет сделать предположение о том, что предлагаемый алгоритм способен выходить из локальных оптимумов.

На рис. 1 представлены две диаграммы «Глобальный оптимум» и «Текущий оптимум» оси абсцисс которых соответствуют итерациям разработанного алгоритма. Значения оси ординат диаграммы «Глобальный оптимум» соответствуют стоимости наилучшего решения найденного алгоритмом за все предшествующие итерации. По оси ординат диаграммы «Текущий оптимум» откладывается стоимость наилучшего решения найденного на текущей итерации. На 4 и 5 итерациях рис. 1 диаграммы «Текущий оптимум» алгоритм покидает зону локального оптимума, но затем, на 6–9 итерациях определяет более предпочтительное решение. Данный факт позволяет сделать предположение о том, что предлагаемый алгоритм способен выходить из локальных оптимумов.

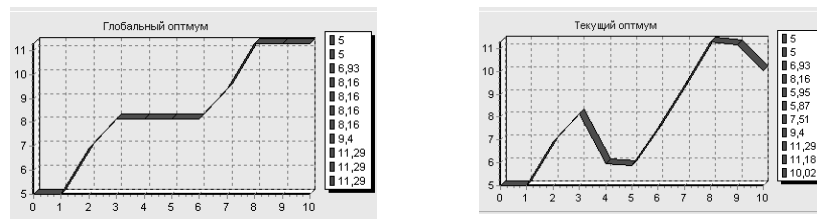


Рис. 1. Распределение результатов работы алгоритма по итерациям

На рис. 2 ось абсцисс соответствует размерности решаемой задачи (N), а ось ординат – погрешности алгоритма. Из рис. 2 следует, что несмотря на всплески на 3 и 7 итерациях, в целом, лавинообразное накопление погрешности отсутствует. Данное утверждение позволяет сделать предположение о том, что рост размерностей решаемых задач не требует пропорционального увеличения вычислительных ресурсов необходимых для получения приемлемого решения. Исследуем рост погрешности алгоритма на размерностях 1–10, приравняв число итераций алгоритма и число особей роя 5. Графическое представление динамики изменения погрешности в зависимости от размерности задачи приведено на рис. 2.

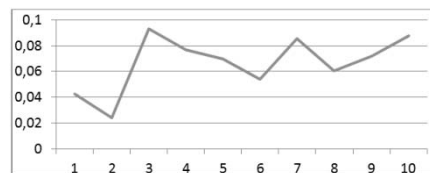


Рис. 2. Зависимость погрешности алгоритма от размерности задачи

Рассмотрим ВСА на задачах размерности 5–100. На рис. 3 графически представлена зависимость времени работы алгоритма от размерности решаемых задач.

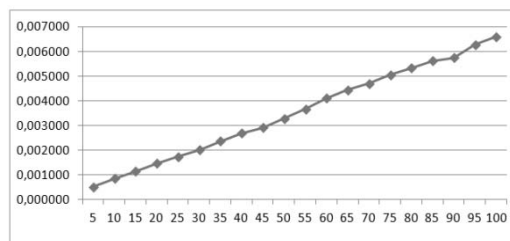


Рис. 3. ВСА на задачах размерности 5–100

Из графика, представленного на рис. 3, можно сделать вывод, что вычислительная сложность представленного алгоритма составляет $O(n)$. На рис. 4 представлена зависимость погрешности алгоритма от размерности задачи $N=10-100$.

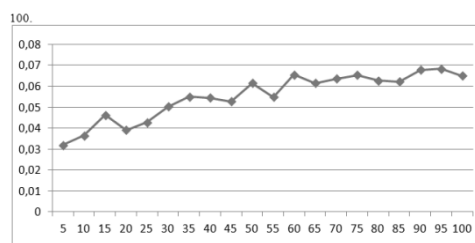


Рис. 4. Погрешности алгоритма на задачах размерности 5–100

На основании данных представленных рис. 3, 4 можно заключить, что предлагаемый алгоритм способен решать поставленные задачи за линейное время.

Заключение. На основании полученных экспериментальных данных, частично представленных на рисунках (см. рис. 1–4) можно заключить, что на задачах размерности 1–100 предложенный алгоритм способен определять субоптимальные решения, погрешность которых не превышает 10 %. При этом вычислительная сложность алгоритма близка к линейной.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик В.М., Комар А.В. Особенности построения синергетических систем управления бионическим поиском в задачах размещения // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 8-14.
2. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. Монография. – М.: Физматлит, 2006.
3. Чернышев Ю.О., Венцов Н.Н., Крупенин А.В., Щербаков А.В., Мухтаров С.А. Адаптивная оптимизация запросов в современных системах управления базами данных / Монография. – Краснодар: ФВАС, 2011.
4. Курейчик В.М. Биоинспирированный поиск с использованием сценарного подхода // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 7-12.
5. Лебедев Б.К., Лебедев В.Б. Поисковые процедуры канальной трассировки, базирующиеся на моделировании адаптивного поведения роя частиц в пространстве решений с неупорядоченным лингвистическим шкалированием // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 15-22.
6. Dorigo M. and Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
7. Eberhart R., Kennedy J. A New Optimizer using Particle Swarm Theory // In Proceedings of the Sixth International Symposium on Micro machine and Human Science 1995. – P. 39-43.

8. *Engelbrecht A.* Computational intelligence: an introduction – John Wiley and Sons Ltd., 2007. – 597 p.
9. *Abraham A., Grosan G.* Swarm Intelligence in Data Mining –Springer, 2006. – 267 p.

Статью рекомендовал к опубликованию к.т.н., доцент Е.В. Нужнов.

Чернышев Юрий Олегович – Донской государственный технический университет; e-mail: myvnn@list.ru; 344000, г. Ростов-на-Дону, пл. Гагарина, 1; тел.: 88632738582; кафедра автоматизации производственных процессов; д.т.н.; профессор.

Венцов Николай Николаевич – e-mail: vencov@list.ru; кафедра информационных технологий; к.т.н.; доцент.

Панасенко Павел Александрович – Филиал Военной академии связи (г. Краснодар); e-mail: vasandevg@mail.ru; г Краснодар, ул. Красина, 4; тел.: +78612683509; соискатель.

Chernyshev Yury Olegovich – Don State Technical University, Russia, e-mail: myvnn@list.ru; 1, Gagarin street, Rostov-on-Don, Russia; phone: +78632738582; the department of automation of productions; the department of automation of productions; dr. of eng. sc.; professor

Vencov Nikolay Nikolaevich – Don State Technical University; e-mail: vencov@list.ru; the department of information technologies; cand. of eng. sc.; associate professor.

Panasenko Pavel Alexandrovich – Branch of Military academy of communication (Krasnodar); e-mail: vasandevg@mail.ru; 4, Krasina street, Krasnodar, Russia; phone: +78612683509; competitor.