

УДК 621.396.96

ЭВРИСТИЧЕСКАЯ ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ АНТЕНН RFID-СИСТЕМЫ ПРИ РЕАЛИЗАЦИИ КОМБИНИРОВАННОГО МЕТОДА ПРОСТРАНСТВЕННОЙ ДВУМЕРНОЙ ЛОКАЛИЗАЦИИ

Ю.Б. ГИМПЛЕВИЧ, Д.А. САВОЧКИН

*Севастопольский национальный технический университет
Университетская, 33, Севастополь, 99053, Украина*

Поступила в редакцию 14 января 2015

Проведена адаптация четырех эвристических алгоритмов оптимизации (генетического алгоритма, алгоритма роста растений, алгоритма имитации отжига, алгоритма роя частиц) к задаче поиска наилучшего размещения антенн при использовании комбинированного метода пространственной двумерной RFID-локализации. В ходе эксперимента наилучший вариант размещения антенн получен с помощью алгоритма имитации отжига.

Ключевые слова: RFID, локализация, эвристическая оптимизация, размещение антенн, генетический алгоритм, алгоритм роста растений, алгоритм имитации отжига, алгоритм роя частиц.

Введение

С целью определения местоположения объектов внутри закрытых помещений часто используют системы на базе технологии радиочастотной идентификации (radio frequency identification, RFID) [1]. При использовании таких систем двумерная оценка местоположения объектов рассчитывается путем обработки измерительной информации, которую получают с помощью нескольких антенн от RFID-меток, установленных на объектах. Для повышения точности локализации при вычислении оценок местоположения могут использоваться комбинированные методы локализации, включающие в себя несколько алгоритмов. Каждый из таких алгоритмов формирует независимую оценку местоположения объекта, после чего вычисляется общая оценка в виде среднего от сформированных оценок [2].

Важным вопросом при создании систем пространственной локализации является поиск такого размещения антенн, которое позволяет достичь максимальной точности локализации [3–4]. При этом для каждой антенны ищется точка расположения в двумерном пространстве размещения антенн (обычно на потолке помещения) и угол поворота (при использовании направленных антенн). В настоящей работе проводится адаптация известных эвристических алгоритмов оптимизации к проблеме размещения антенн при реализации комбинированного метода локализации, а также количественное сравнение эффективности алгоритмов.

Критерий оптимальности размещения антенн

Рассматриваемый в работе комбинированный метод локализации включает в себя три базовых алгоритма: алгоритм трилатерации, алгоритм отпечатков и алгоритм пересечений [2]. Целевая функция $S(a)$ оптимизации размещения антенн вычисляется с учетом факторов, характеризующих оптимальность размещения при реализации базовых алгоритмов. Критерием оптимальности является максимум целевой функции $S(a)$, задаваемой в виде выражения

$$S(a) = c_1 a^2 c_2 a^3 \left[w_{\text{трил}} a \overline{S_{\text{трил}}(a)} + w_{\text{отп}} a \overline{S_{\text{отп}}(a)} + w_{\text{пер}} a \overline{S_{\text{пер}}(a)} \right], \quad (1)$$

где a – вектор координат точек размещения и величин поворотов антенн; $c'_1(a)$ – отношение площади той части центральной области зоны локализации, что покрыта зонами действия хотя бы одной антенны, к общей площади центральной области зоны локализации; $c_1(a)$ – отношение площади той части зоны локализации, что покрыта зонами действия хотя бы одной антенны, к общей площади зоны локализации; $w_{\text{трил}}(a)$, $w_{\text{отп}}(a)$, $w_{\text{пер}}(a)$ – весовые коэффициенты алгоритмов трилатерации, отпечатков и пересечений, соответственно; $S_{\text{трил}} a$, $S_{\text{отп}} a$, $S_{\text{пер}} a$ – нормированные величины факторов $S_{\text{трил}}(a)$, $S_{\text{отп}}(a)$, $S_{\text{пер}}(a)$, характеризующих оптимальность размещения антенн при реализации алгоритмов трилатерации, отпечатков и пересечений, соответственно (нормирование проводится с целью приведения к общему диапазону допустимых значений).

Весовые коэффициенты $w_{\text{трил}}(a)$, $w_{\text{отп}}(a)$, $w_{\text{пер}}(a)$ задаются следующим образом:

$$\begin{cases} w_{\text{трил}} a = c_3 a / 3; \\ w_{\text{отп}} a = w_{\text{пер}} a = \frac{1}{3} \left[1 + \frac{1 - c_3 a}{2} \right], \end{cases}$$

где $c_3(a)$ – отношение площади той части зоны локализации, что покрыта зонами действия трех или более антенн, к общей площади зоны локализации.

В качестве фактора $S_{\text{трил}}(a)$ используется усредненная величина коэффициента горизонтального снижения точности (horizontal dilution of precision, HDOP):

$$S_{\text{трил}} a = \frac{1}{|L_3 a|} \int_{l \in L_3 a} h a, l dl,$$

где $|L_3(a)|$ – площадь той части зоны локализации, что покрыта зонами действия трех или более антенн; $L_3(a)$ – множество точек той части зоны локализации, что покрыта зонами действия трех или более антенн; $h(a, l)$ – величина HDOP для точки l [5].

В качестве фактора $S_{\text{отп}}(a)$ используется коэффициент покрытия зоны локализации:

$$S_{\text{отп}}(a) = \frac{1}{|L|} \int_{l \in L} c a, l dl,$$

где $|L|$ – площадь зоны локализации; L – множество точек зоны локализации; $c(a, l)$ – число зон действия, внутри которых находится точка l при размещении антенн согласно вектору a .

В качестве фактора $S_{\text{пер}}(a)$ используется величина средней ошибки локализации, обеспечиваемой алгоритмом пересечений:

$$S_{\text{пер}}(a) = \frac{1}{|L|} \int_{l \in L} e(a, l) dl,$$

где $e(a, l)$ – ошибка локализации для точки l при размещении антенн согласно вектору a , которая определяется как расстояние между точкой l и оценкой местоположения RFID-метки, формируемой алгоритмом пересечений при расположении метки в точке l [6].

Эвристические алгоритмы оптимизации

Высокая степень нелинейности целевой функции (1) не позволяет аналитически сформировать выражение для нахождения ее максимума. Вариант полного перебора всех возможных значений a также является невозможным, поскольку при необходимости размещения достаточно большого числа антенн вычислительная сложность такого перебора будет крайне велика. В связи с этим мы предлагаем использовать эвристические алгоритмы оптимизации. В данном разделе мы приводим краткое описание четырех алгоритмов оптимизации (генетического алгоритма, алгоритма роста растений, алгоритма имитации отжига, алгоритма роя частиц) и описываем процесс их адаптации к нашей задаче.

Генетический алгоритм оптимизации (genetic algorithm, GA) функционирует согласно принципам эволюции в животном мире [7]. В ходе работы алгоритма вначале создается популяция из P особей (решений задачи оптимизации) со случайными параметрами. При случайном создании особей учитываются два условия: антенны не должны находиться ближе, чем на 50 см друг к другу; зоны действия антенн должны покрывать минимум 85 % зоны локализации. С помощью этих условий мы сужаем начальное пространство решений путем исключения заведомо неоптимальных вариантов для увеличения быстродействия алгоритма.

Далее для каждой из особей рассчитывается значение функции приспособленности (целевой функции решения). После этого выбирается E особей с наибольшими значениями функции, называемые элитами. Все элиты переходят в следующую эпоху (итерацию цикла оптимизации). Кроме этого выбранная случайным образом часть элит мутирует (происходит случайное изменение нескольких параметров соответствующих решений). Сформированные в результате мутации M особей также переходят в следующую эпоху. Помимо этого часть элит скрещивается между собой (формируются новые особи, у которых часть параметров соответствует параметрам одного из родительских особей, а часть – параметрам другого). Сформированные при этом C особей также переходят в следующую эпоху. Все остальные особи популяции отмирают (соответствующие решения задачи оптимизации удаляются из рассмотрения). В следующей эпохе заново формируется $P - E - M - C$ новых особей и процесс эволюции (поиска решений) продолжается дальше, пока не будет достигнут критерий останова (в настоящей работе – по прошествии восьми часов). Результатом работы алгоритма считается то решение, для которого было получено максимальное значение целевой функции за все время работы. В нашей реализации алгоритма мы использовали следующие параметры: общий размер популяции $P = 20$; количество элит в популяции $E = 4$; количество мутирующих особей в популяции $M = 5$; количество особей формируемых в ходе скрещивания $C = 2$.

Процедура мутации особи для нашей проблемы размещения антенн была сформирована следующим образом. Для каждой из антенн выполняются поочередно следующие действия:

1) с вероятностью 0,5 делается 10 попыток изменить позицию антенны по каждой из координат (x и y) на случайную величину, распределенную нормально с нулевым математическим ожиданием и среднеквадратическим отклонением равным 4 см. Попытка считается успешной, если в результирующем размещении все антенны находятся на расстоянии как минимум 50 см друг от друга. Если хотя бы одна попытка успешна, то дальнейшие попытки не проводятся;

2) с вероятностью 0,5 изменяется величина φ поворота антенны на случайную величину, распределенную нормально с нулевым математическим ожиданием и среднеквадратическим отклонением 9° .

Процедура скрещивания двух особей (a_1 и a_2) формулируется следующим образом. Создается полная копия особи a_1 : $a_c \leftarrow a_1$. Далее в зоне локализации случайным образом выбирается некоторая точка. Для каждой из особей a_2 и a_c находится по одной антенне, расположенной ближе всего к этой точке. После этого позиция и величина поворота найденной антенны особи a_c заменяются позицией и величиной поворота найденной антенны особи a_2 . Результатом процедуры скрещивания является особь a_c .

Алгоритм роста растений (plant growth algorithm, PGA) базируется на принципе развития растений [8]. В ходе работы алгоритма вначале формируется одно случайное решение задачи оптимизации (в терминологии алгоритма – корень), которое записывается в массив решений (при этом учитываются два условия формирования размещений антенн, представленные в описании генетического алгоритма). Далее запускается цикл поиска оптимального решения. На каждой итерации цикла выбирается одно из уже имеющихся в массиве решений. Вероятность p_i выбора i -го решения a_i при этом определяется как

$$p_i = \frac{S(a_i) - \beta \cdot \min_{j \in J} [S(a_j)]}{\sum_{k \in J} \left\{ S(a_k) - \beta \cdot \min_{l \in J} [S(a_l)] \right\}},$$

где β – коэффициент, позволяющий задать степень, с которой уменьшаются величины целевой функции относительно минимального значения в массиве решений (мы задали его равным $\beta = 0,7$); J – множество номеров решений содержащихся в массиве решений.

После выбора одного из решений создается Q его копий, называемых листьями. Каждый лист случайным образом видоизменяется (принцип видоизменения аналогичен процедуре мутации генетического алгоритма), после чего для него считается величина целевой функции. Далее все листья, у которых величина целевой функции больше величины целевой функции того решения, которое использовалось как прообраз, добавляются в массив решений. В конце итерации из массива решений стираются все решения кроме B решений с наибольшими значениями целевой функции. Результатом работы алгоритма считается то решение, для которого было получено максимальное значение целевой функции за все время работы. В нашей реализации алгоритма мы использовали следующие параметры: количество листьев на каждой итерации $Q = 5$; количество оставляемых решений в массиве решений $B = 5$.

Алгоритм имитации отжига (simulated annealing algorithm, SAA) основан на физическом процессе, происходящем при отжиге вещества [9]. В ходе работы алгоритма первоначально формируется одно случайное решение задачи (рабочее решение) с учетом двух условий, представленных в описании генетического алгоритма, а также задается начальная температура T . Далее запускается цикл поиска оптимального решения. На каждой итерации цикла имеющееся решение задачи случайным образом видоизменяется (принцип видоизменения аналогичен процедуре мутации генетического алгоритма), после чего для него рассчитывается величина целевой функции. Если величина целевой функции нового решения оказывается выше предыдущего, то новое решение используется далее в качестве рабочего. Если же величина целевой функции оказывается меньше, то новое решение также может быть выбрано в качестве рабочего, но с определенной вероятностью r_i , которая вычисляется как

$$r_i = \exp\{[S(a_i) - S(a_{i-1})]/T\},$$

где $S(a_i)$, $S(a_{i-1})$ – величины целевой функции нового и рабочего решений, соответственно.

В конце итерации цикла температура T снижается путем умножения на некоторый коэффициент охлаждения Y . Результатом работы алгоритма считается последнее рабочее решение. В качестве дополнительной меры решения проблемы выхода из локальных экстремумов использовалась процедура возвращения решения назад в случае отсутствия улучшения в течение V итераций. В своей реализации алгоритма авторы использовали следующие параметры: начальная температура $T = 1$; коэффициент охлаждения $Y = 0,992$; число итераций без улучшения, после которых происходит возвращение, $V = 10$.

Алгоритм роя частиц (particle swarm algorithm, PSA) основан на логике поведения стай птиц или косяков рыбы [10]. В ходе работы алгоритма первоначально создается K случайных решений задачи оптимизации (частиц в терминологии алгоритма) с учетом двух условий представленных в описании генетического алгоритма. Для каждой частицы случайным образом задается ее начальная скорость в пространстве решений в виде вектора скорости. В нашем случае вектор скорости содержит N элементов (N – число антенн системы), где в каждом элементе содержатся компоненты скорости изменения позиции антенны по x и y координатам, а также скорость вращения антенны. После создания частиц и задания их начальных скоростей запускается цикл поиска оптимального решения. На каждой итерации цикла проводится перерасчет скоростей частиц, после чего частицы перемещаются в пространстве решений на расстояние, определяемое с учетом их скоростей. В конце каждой итерации для каждой частицы вычисляется величина целевой функции. В качестве результата работы алгоритма выдается наилучший результат, полученный за все время среди всех частиц.

Вектор $V_{i,j}$ скорости i -й частицы на j -й итерации вычисляется согласно выражению

$$V_{i,j} = \omega V_{i,j-1} + 2 \cdot U(0,1) \cdot \{a_{i,best} - a_{i,j}\} + 2 \cdot U(0,1) \cdot \{a_{best} - a_{i,j}\}$$

где $\omega = 0,9$ – коэффициент инерции, постепенно уменьшающийся на каждой итерации, пока не достигнет значения $0,2$; $V_{i,j-1}$ – вектор скорости i -й частицы на $(j-1)$ -й итерации; $U(0, 1)$ – случайная величина, распределенная равномерно от 0 до 1 ; $a_{i,best}$ – наилучшая позиция (вектор размещения антенн) i -й частицы за все время работы алгоритма до j -й итерации; a_{best} –

наилучшая позиция среди всех частиц за все время работы алгоритма до j -й итерации; a_{ij} – позиция i -й частицы на j -й итерации.

Расчет новой позиции a_{ij} i -й частицы на j -й итерации проводится путем сложения вектора $a_{i,j-1}$ i -й частицы на предыдущей итерации с рассчитанным вектором скорости V_{ij} . При этом в случае выхода значений позиций антенн за пределы зоны локализации (отрицательные значения или значения большие размера зоны) мы выполняли возврат значений в допустимую область, а также меняли знак соответствующей компоненты вектора скорости.

Экспериментальная часть

Мы провели поиск оптимального размещения 16 антенн RFID-системы с помощью четырех описанных в предыдущем разделе эвристических алгоритмов. Поиск размещения проводился для зоны локализации с размером $5 \text{ м} \times 5 \text{ м}$. Зоны действия антенн задавались в виде эллипсов с размерами малой и большой осей равными 1,6 м и 2,5 м соответственно. Зоны неуверенного приема задавались в виде эллипсов с размерами осей равными 1,9 м и 3 м (под зоной неуверенного приема подразумевается такая зона, в пределах которой RFID-метки отвечают антенне с вероятностью большей нуля, но меньшей единицы).

Вычисления проводились на компьютере с процессором Intel Core i5 с частотой 1,8 ГГц и объемом оперативной памяти 4 Гб. Расчет значений HDOP, коэффициента покрытия и ошибки локализации алгоритма пересечений (для последующего их использования при вычислении значений целевой функции (1)) проводился в точках зоны локализации с шагом в 2 см. При этом каждый алгоритм запускался по 10 раз на 8 часов. Такое время работы было выбрано экспериментально, после того как было замечено, что для всех четырех алгоритмов дальнейшая их работа практически не улучшает результат. Кроме того, данное значение было выбрано и из чисто практических соображений: общее время вычислений заняло более 13 суток, и дальнейшее увеличение времени работы было сочтено нецелесообразным.

После нахождения алгоритмами оптимальных векторов размещения антенн и соответствующих им значений целевой функции были вычислены основные параметры распределения величин целевой функции для каждого алгоритма (см. таблицу).

Параметры распределения величин целевой функции, полученные в ходе десятикратного выполнения алгоритмов оптимизации

Алгоритм оптимизации	Величина целевой функции			
	Наилучшее значение	Наихудшее значение	Среднее значение	Стандартное отклонение
GA	0,5129	0,5023	0,5080	0,003635
PSA	0,4865	0,4666	0,4759	0,007917
SAA	0,5277	0,5139	0,5179	0,004060
PGA	0,5143	0,4989	0,5088	0,005659

На рис. 1 представлен график процесса работы алгоритмов (зависимость величины целевой функции $S(a)$ от времени работы t алгоритма). Значения на графике сформированы путем усреднения величин, полученных в ходе 10 запусков алгоритмов.

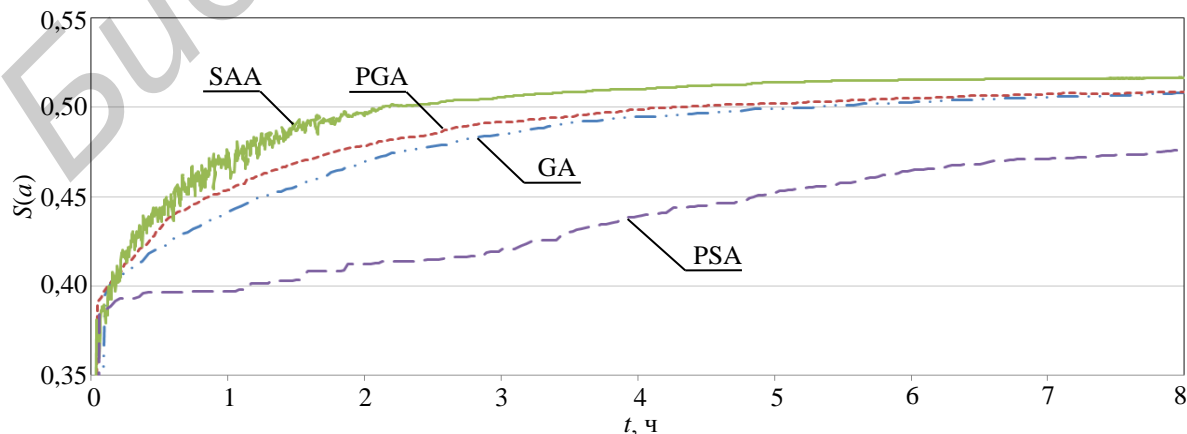


Рис. 1. Зависимость величины целевой функции $S(a)$ от времени t работы алгоритма

На рис. 2 представлены наилучшие варианты размещения антенн, полученные каждым из алгоритмов оптимизации (величины целевой функции для этих вариантов представлены во втором столбце таблицы).

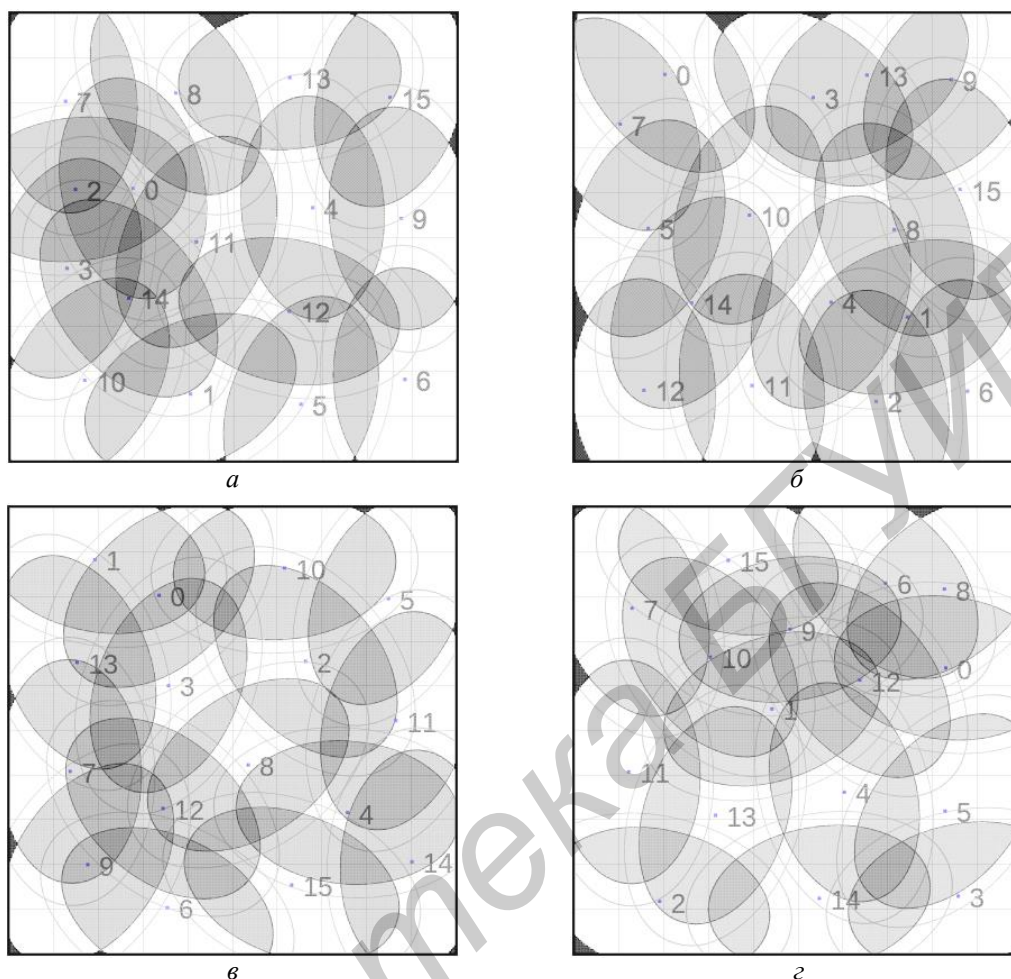


Рис. 2. Наилучшие варианты размещения антенн, полученные в ходе работы алгоритмов GA (а), PSA (б), SAA (в), PGA (г)

По результатам, представленным в таблице и на рис. 1, видно, что наилучшее размещение антенн (величина целевой функции 0,5277) получено алгоритмом имитации отжига, в то время как наихудшее (величина целевой функции 0,4865) получено при использовании алгоритма роя частиц. Наиболее стабильные (среди 10 перезапусков) результаты получены генетическим алгоритмом: величина стандартного отклонения равна 0,003635.

Высокую эффективность алгоритма имитации отжига можно объяснить тем, что он последовательно пытается улучшить единственное рабочее решение, в то время как все остальные алгоритмы постоянно работают с некоторым массивом решений, что очевидно существенно замедляет их работу. Так, за 8 часов работы алгоритм имитации отжига успевал выполнить около 3300 итераций, в то время как все остальные алгоритмы выполняли за это же время от 200 до 500 итераций. Этот эффект нельзя считать недостатком алгоритмов GA, PGA и PSA, поскольку их работа сразу над несколькими решениями должна помогать в преодолении проблемы выхода из локальных экстремумов. Можно было бы даже ожидать, что эти алгоритмы решат такую проблему успешнее, чем алгоритм имитации отжига (который может выйти из локального экстремума только на ранних итерациях, пока величина температуры T еще достаточно высока). Однако был достигнут обратный результат. Авторы считают, что это произошло по той причине, что использованная процедура видоизменения решений на каждой итерации (мутация в терминологии генетического алгоритма) предполагает возможность существенного (хоть и маловероятного) изменения позиций антенн за раз. При этом появляется

возможность выйти сколь угодно далеко из любого локального экстремума. Вероятно, благодаря этому алгоритм имитации отжига и работал достаточно быстро и при том эффективно.

Кроме этого, интересен тот факт, что алгоритм роя частиц показал в рассматриваемом случае самый плохой результат, хотя в литературе он описывается как достаточно эффективный. Авторы полагают, что это произошло по той причине, что во всех остальных алгоритмах при видоизменении решений использовалось два условия: нахождения антенн не ближе чем на 50 см друг к другу; требования покрытия как минимум 85 % зоны локализации зонами действия антенн. Эти условия позволяли не выполнять трудозатратную (время выполнения до 15 секунд) операцию по вычислению величины целевой функции для тех решений, которые заведомо неоптимальны. В алгоритм роя частиц не удалось привести такие условия из-за его принципа работы, предполагающего использование скоростей изменения частиц (решений задачи оптимизации). В связи с этим существенная часть времени работы алгоритма тратилась на расчет целевой функции для заведомо плохих вариантов размещения антенн. Возможно, если каким-то образом разработать реализацию алгоритма роя частиц, совмещающую его классический принцип работы с вышеприведенными условиями, то можно будет ожидать повышения его эффективности.

Заключение

Разработаны реализации четырех эвристических алгоритмов оптимизации для решения задачи размещения антенн RFID-системы при использовании комбинированного метода двумерной пространственной локализации. В ходе экспериментальных исследований наилучшее размещение 16 антенн в зоне локализации размером 5 м×5 м получено с помощью алгоритма имитации отжига. В будущем планируется сравнить эффективность рассмотренных в работе алгоритмов с некоторыми другими эвристическими алгоритмами (муравьиный алгоритм, алгоритм бактерий и др.).

HEURISTIC OPTIMIZATION OF ANTENNA DEPLOYMENT FOR COMBINATIONAL TWO-DIMENSIONAL RFID-BASED LOCALIZATION METHOD

Yu.B. GIMPILEVICH, D.A. SAVOCHKIN

Abstract

Four heuristic optimization algorithms (genetic algorithm, plant growth algorithm, simulated annealing algorithm, particle swarm algorithm) are adapted to the problem of the search for the best antenna deployment using a combinational RFID-based two-dimensional localization method. During the experiment the best antenna deployment was obtained by the simulated annealing algorithm.

Список литературы

1. Zhou J., Shi J. // Journal of intelligent manufacturing. 2009. Vol. 20, № 6. P. 695–707.
2. Savochkin D.A. // MIKON 2014. Gdansk, 16–18 June 2014. P. 563–566.
3. Ji Y., Biaz S., Wu S. et al. // ICCCN. Honolulu, 13–16 August 2007. P. 251–256.
4. Fang S.H., Lin T.-N. // WCNC. Sydney, 18–21 April 2010. P. 1–4.
5. Levanon N. // IEE Proceedings - radar, sonar and navigation. 2000. Vol. 147, № 3. P. 149–155.
6. Savochkin D.A. // Матер. конф. «РТ-2014». Севастополь, 12–17 мая 2014. С. 70.
7. Guan Q., Liu Y., Yang Y. et al. // ISDA'06. Jinan, 16–18 October 2006. Vol. 2. P. 567–572.
8. Yihua H., Shilei L. // ICCM. Seoul, 24–26 April 2012. Vol. 1. P. 196–201.
9. Farkas K., Huszak A., Godor G. // Informatics & IT today. 2013. Vol. 1, № 1. P. 28–33.
10. Gong Y.J., Shen M., Zhang J. et al. // IEEE TII. 2012. Vol. 8, № 4. P. 900–912.